

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-353085

(43)Date of publication of application : 24.12.1999

(51)Int.Cl.

G06F 3/00

(21)Application number : 11-137046

(71)Applicant : SUN MICROSYST INC

(22)Date of filing : 18.05.1999

(72)Inventor : LUDOLPH FRANK E  
THARAKAN GEORGE

(30)Priority

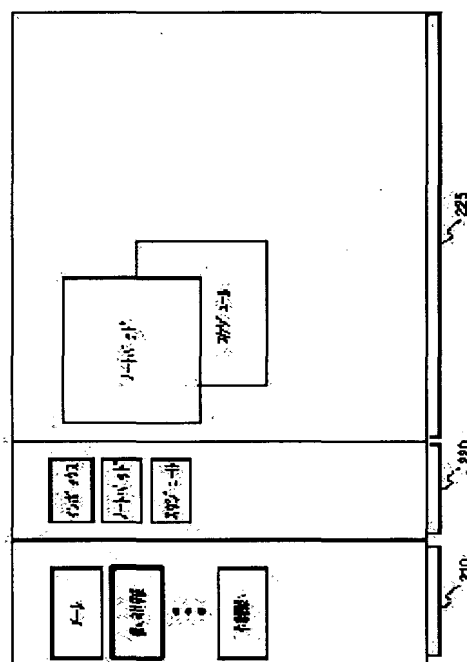
Priority number : 98 85456    Priority date : 28.05.1998    Priority country : US

## (54) WINDOW DISPLAY MANAGEMENT METHOD, WINDOW DISPLAY MANAGEMENT DEVICE AND COMPUTER READABLE RECORDING MEDIUM

(57)Abstract:

**PROBLEM TO BE SOLVED:** To enable a user to easily select and process a window inside a user interface for supporting plural work areas by permitting shared access to the window of a task during execution.

**SOLUTION:** A selector 210 functions as a work area changeover switch and the user comes and goes from/to the different work areas by using it. The user selects the work area by clicking one of icons inside the selector 210 and a computer displays the window held in the work area in response to it. The selector 210 holds the icons corresponding to the plural work areas. The work area for constitution for an individual is activated and it is indicated by a thick line around the icon inside the selector 210. The work area for the constitution for the individual holds the two windows, one corresponds to a notepad task and the other one corresponds to a schedule task.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]



[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

α)

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-353085

(43) 公開日 平成11年(1999)12月24日

(51) Int.Cl.<sup>6</sup>

G 0 6 F 3/00

識別記号

6 5 5

F I

G 0 6 F 3/00

6 5 5 A

審査請求 未請求 請求項の数1 O L 外国語出願 (全 48 頁)

(21) 出願番号 特願平11-137046

(22) 出願日 平成11年(1999) 5月18日

(31) 優先権主張番号 0 9 / 0 8 5 4 5 6

(32) 優先日 1998年 5月28日

(33) 優先権主張国 米国 (U S)

(71) 出願人 598015419

サンマイクロシステムズ インコーポレー  
テッド

アメリカ合衆国、94043 カリフォルニア  
州、マウンテン ビュー、ガルシア アヴ  
ェニュー 2550

(72) 発明者 フランク イー ルドルフ

アメリカ合衆国 94025 カリフォルニア  
州、メンロパーク、エンパーシティー ド  
ライブ 241

(74) 代理人 弁理士 上野 登 (外1名)

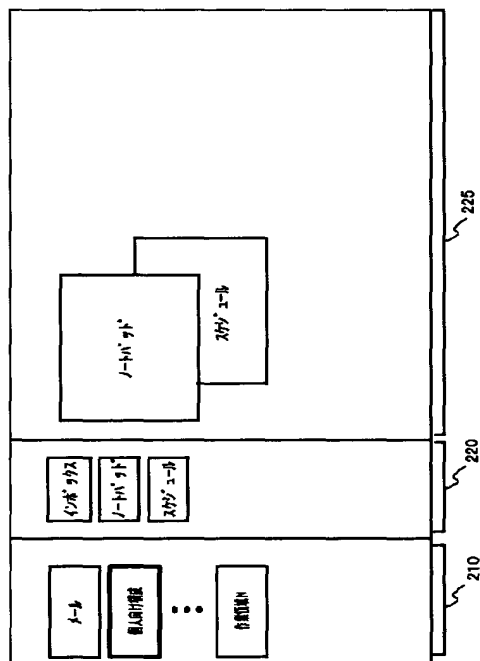
最終頁に続く

(54) 【発明の名称】 ウィンドウディスプレイ管理方法、ウィンドウディスプレイ管理装置、コンピュータ読取可能な記録媒体

(57) 【要約】

【課題】複数の作業領域をサポートするユーザインタフェース内のウィンドウを、簡単にユーザが選択でき処理することができるウィンドウディスプレイ管理方法及びウィンドウディスプレイ管理装置システムを提供すること。

【解決手段】作業領域内にウィンドウが存在しているか否かに関わらず、ウィンドウ内にオープンされた各タスクを表すアイコンを保持するスライド可能なウィンドウパネルを使用する。ユーザは、その作業領域内のウィンドウを作動させ、終了させ、隠蔽し、又は、そのサイズを変更するために、そのスライド可能なウィンドウパネルを使用すればよい。



**【特許請求の範囲】**

【請求項1】 コンピュータによって実行可能なタスクに対応するウィンドウディスプレイを管理するウィンドウディスプレイ管理方法において、  
実行中タスクに対応する複数のウィンドウを表示することができる作業領域を複数提供する複数作業領域提供工程と、

前記実行中タスクに対応するアイコンを含むウィンドウパネルを表示するウィンドウパネル表示工程と、  
前記ウィンドウパネルに表示された前記実行中タスクに対応するアイコンが選択された時に、当該実行中タスクのウィンドウに共有アクセスを許可する共有アクセス許可工程とを含むことを特徴とするウィンドウディスプレイ管理方法。

【請求項2】 更に、ウィンドウ内にオープンされた各タスク用のアイコンを前記ウィンドウパネルに追加するアイコン追加工程を含むことを特徴とする請求項1に記載されるウィンドウディスプレイ管理方法。

【請求項3】 更に、ユーザからタスクを作動させるためのコマンドを受信する作動コマンド受信工程と、  
当該タスク用のウィンドウを生成するウィンドウ生成工程とを含むことを特徴とする請求項1に記載されるウィンドウディスプレイ管理方法。

【請求項4】 更に、ユーザから被選択作業領域内のウィンドウを隠蔽するためのコマンドを受信する隠蔽コマンド受信工程と、  
前記被選択作業領域からのみ前記ウィンドウを除去する限定ウィンドウ除去工程と、  
前記ウィンドウパネル内に前記ウィンドウを表すアイコンを維持するアイコン維持工程とを含むことを特徴とする請求項1に記載されるウィンドウディスプレイ管理方法。

【請求項5】 更に、前記被選択作業領域内の前記ウィンドウのサイズと位置とを記録する記録工程を含むことを特徴とする請求項4に記載されるウィンドウディスプレイ管理方法。

【請求項6】 更に、特定の作業領域内で、ユーザから前記ウィンドウパネル内のアイコンを選択するためのコマンドを受信する選択コマンド受信工程と、  
その被選択アイコンに対応するウィンドウを前記作業領域の前面に表示する前面ウィンドウ表示工程とを含むことを特徴とする請求項1に記載されるウィンドウディスプレイ管理方法。

【請求項7】 更に、ユーザからウィンドウを終了するためのコマンドを受信する終了コマンド受信工程と、  
前記ウィンドウを当該ウィンドウが表示されている各作業領域から除去する非限定ウィンドウ除去工程と、  
その終了ウィンドウに対応するアイコンを前記ウィンドウパネルから削除するアイコン削除工程とを含むことを特徴とする請求項1に記載されるウィンドウディスプレイ

管理方法。

【請求項8】 更に、複数の作業領域内にウィンドウを表示するウィンドウ表示工程と、  
他の作業領域内のウィンドウのサイズに影響を与えることなく、ある作業領域内のウィンドウのサイズを変更するサイズ変更工程とを含むことを特徴とする請求項1に記載されるウィンドウディスプレイ管理方法。

【請求項9】 更に、複数の作業領域内にウィンドウを表示するウィンドウ表示工程と、  
他の作業領域内のウィンドウの位置に影響を与えることなく、ある作業領域内のウィンドウの配置を変更する配置変更工程とを含むことを特徴とする請求項1に記載されるウィンドウディスプレイ管理方法。

【請求項10】 更に、作業領域切替スイッチの後ろに前記ウィンドウパネルを隠蔽する隠蔽工程と、  
ユーザから前記作業領域切替スイッチを選択するためのコマンドを受信する選択コマンド受信工程と、  
前記スイッチを選択するユーザコマンドに応答して前記ウィンドウパネルを表示するウィンドウパネル表示工程とを含むことを特徴とする請求項1に記載されるウィンドウディスプレイ管理方法。

【請求項11】 コンピュータによって実行可能なタスクに対応するウィンドウディスプレイを管理するウィンドウディスプレイ管理手順であって、  
実行中タスクに対応する複数のウィンドウを表示することができる作業領域を複数提供する複数作業領域提供手順と、

前記実行中タスクに対応するアイコンを含むウィンドウパネルを表示するウィンドウパネル表示手順と、  
前記ウィンドウパネルに表示された前記実行中タスクに対応するアイコンが選択された時に、当該実行中タスクのウィンドウに共有アクセスを許可する共有アクセス許可手順と、  
を含むウィンドウディスプレイ管理手順をコンピュータに実行させるための命令を記録したコンピュータ読取可能な記録媒体。

【請求項12】 更に、ウィンドウ内にオープンされた各タスク用のアイコンを前記ウィンドウパネルに追加するアイコン追加手順を含むウィンドウディスプレイ管理手順をコンピュータに実行させるための命令を記録した請求項11に記載されるコンピュータ読取可能な記録媒体。

【請求項13】 更に、ユーザからタスクを作動させるためのコマンドを受信する作動コマンド受信手順と、  
前記タスク用のウィンドウを生成するウィンドウ生成手順とを含むウィンドウディスプレイ管理手順をコンピュータに実行させるための命令を記録した請求項11に記載されるコンピュータ読取可能な記録媒体。

【請求項14】 更に、ユーザから被選択作業領域内のウィンドウを隠蔽するためのコマンドを受信する隠蔽コ

マンド受信手順と、  
前記被選択作業領域からのみ前記ウインドウを除去する  
限定ウインドウ除去手順と、  
前記ウインドウパネル内に前記ウインドウを表すアイコン  
を維持するアイコン維持手順とを含むウインドウディス  
プレイ管理手順をコンピュータに実行させるための命  
令を記録した請求項11に記載されるコンピュータ読取可  
能な記録媒体。

【請求項15】 更に、前記被選択作業領域内の前記ウ  
インドウのサイズと位置とを記録する記録手順を含むウ  
インドウディスプレイ管理手順をコンピュータに実行さ  
せるための命令を記録した請求項14に記載されるコンピ  
ュータ読取可能な記録媒体。

【請求項16】 更に、特定の作業領域内で、ユーザか  
ら前記ウインドウパネル内のアイコンを選択するための  
コマンドを受信する選択コマンド受信手順と、  
その被選択アイコンに対応するウインドウを前記作業領  
域の前面に表示する前面ウインドウ表示手順とを含むウ  
インドウディスプレイ管理手順をコンピュータに実行さ  
せるための命令を記録した請求項11に記載されるコンピ  
ュータ読取可能な記録媒体。

【請求項17】 更に、ユーザからウインドウを終了す  
るためのコマンドを受信する終了コマンド受信手順と、  
前記ウインドウを当該ウインドウが表示されている各作  
業領域から除去する非限定ウインドウ除去手順と、  
その終了ウインドウに対応するアイコンを前記ウインド  
ウパネルから削除するアイコン削除手順とを含むウイン  
ドウディスプレイ管理手順をコンピュータに実行させる  
ための命令を記録した請求項11に記載されるコンピュ  
ータ読取可能な記録媒体。

【請求項18】 更に、複数の作業領域内にウインドウ  
を表示するウインドウ表示手順と、  
他の作業領域内のウインドウのサイズに影響を与えるこ  
となく、ある作業領域内のウインドウのサイズを変更す  
るサイズ変更手順とを含むウインドウディスプレイ管理  
手順をコンピュータに実行させるための命令を記録した  
請求項11に記載されるコンピュータ読取可能な記録媒  
体。

【請求項19】 更に、複数の作業領域内にウインドウ  
を表示するウインドウ表示工程と、  
他の作業領域内のウインドウの位置に影響を与えること  
なく、ある作業領域内のウインドウの配置を変更する配  
置変更工程とを含むウインドウディスプレイ管理手順を  
コンピュータに実行させるための命令を記録した請求項  
11に記載されるコンピュータ読取可能な記録媒体。

【請求項20】 更に、作業領域切換スイッチの後ろに  
前記ウインドウパネルを隠蔽する隠蔽手順と、  
ユーザから前記作業領域切換スイッチを選択するための  
コマンドを受信する選択コマンド受信手順と、  
前記スイッチを選択するユーザコマンドに応答して前記

ウインドウパネルを表示するウインドウパネル表示手順  
とを含むウインドウディスプレイ管理手順をコンピュ  
ータに実行させるための命令を記録した請求項11に記載さ  
れるコンピュータ読取可能な記録媒体。

【請求項21】 コンピュータによって実行可能なタス  
クに対応するウインドウディスプレイを管理するウイン  
ドウディスプレイ管理装置において、  
プログラム命令を保持したメモリと、  
前記プログラム命令を使用するように構成されたプロセ  
ッサとを備え、当該プロセッサが、  
実行中タスクに対応する複数のウインドウを表示するこ  
とができる作業領域を複数提供する複数作業領域提供手  
段と、  
前記実行中タスクに対応するアイコンを含むウインドウ  
パネルを表示するウインドウパネル表示手段と、  
前記ウインドウパネルに表示された前記実行中タスクに  
対応するアイコンが選択された時に、当該実行中タスク  
のウインドウに共有アクセスを許可する共有アクセス許  
可手段とを備えたことを特徴とするウインドウディスプ  
レイ管理装置。

【請求項22】 更に、前記プロセッサは、  
ウインドウ内にオープンされた各タスク用のアイコンを  
前記ウインドウパネルに追加するアイコン追加手段を備  
えたことを特徴とする請求項21に記載されるウインドウ  
ディスプレイ管理装置。

【請求項23】 更に、前記プロセッサは、  
ユーザからタスクを作動させるためのコマンドを受信す  
る作動コマンド受信手段と、  
前記タスク用のウインドウを生成するウインドウ生成手  
段とを備えたことを特徴とする請求項21に記載されるウ  
インドウディスプレイ管理装置。

【請求項24】 更に、前記プロセッサは、  
ユーザから被選択作業領域内のウインドウを隠蔽するた  
めのコマンドを受信する隠蔽コマンド受信手段と、  
前記被選択作業領域からのみ前記ウインドウを除去する  
限定ウインドウ除去手段と、  
前記ウインドウパネル内に前記ウインドウを表すアイコ  
ンを維持するアイコン維持手段とを備えたことを特徴と  
する請求項21に記載されるウインドウディスプレイ管理  
装置。

【請求項25】 更に、前記プロセッサは、  
前記被選択作業領域内の前記ウインドウのサイズと位置  
とを記録する記録手段を備えたことを特徴とする請求項  
24に記載されるウインドウディスプレイ管理装置。

【請求項26】 更に、前記プロセッサは、  
特定の作業領域内で、ユーザから前記ウインドウパネル  
内のアイコンを選択するためのコマンドを受信する選択  
コマンド受信手段と、  
その被選択アイコンに対応するウインドウを前記作業領  
域の前面に表示する前面ウインドウ表示手段とを備えた

ことを特徴とする請求項21に記載されるウインドウディスプレイ管理装置。

【請求項27】 更に、前記プロセッサは、ユーザからウインドウを終了するためのコマンドを受信する終了コマンド受信手段と、前記ウインドウを当該ウインドウが表示されている各作業領域から除去する非限定ウインドウ除去手段と、その終了ウインドウに対応するアイコンを前記ウインドウパネルから削除するアイコン削除手段とを備えたことを特徴とする請求項21に記載されるウインドウディスプレイ管理装置。

【請求項28】 更に、前記プロセッサは、複数の作業領域内にウインドウを表示するウインドウ表示手段と、他の作業領域内のウインドウのサイズに影響を与えることなく、ある作業領域内のウインドウのサイズを変更するサイズ変更手段とを備えたことを特徴とする請求項21に記載されるウインドウディスプレイ管理装置。

【請求項29】 更に、前記プロセッサは、複数の作業領域内にウインドウを表示するウインドウ表示手段と、他の作業領域内のウインドウの位置に影響を与えることなく、ある作業領域内のウインドウの配置を変更する配置変更手段とを備えたことを特徴とする請求項21に記載されるウインドウディスプレイ管理装置。

【請求項30】 更に、前記プロセッサは、作業領域切換スイッチの後ろに前記ウインドウパネルを隠蔽する隠蔽手段と、ユーザから前記作業領域切換スイッチを選択するコマンドを受信する選択コマンド受信手段と、前記スイッチを選択するユーザコマンドに応答して前記ウインドウパネルを表示するウインドウパネル表示手段とを備えたことを特徴とする請求項21に記載されるウインドウディスプレイ管理装置。

【請求項31】 コンピュータによって実行可能なタスクに対応するウインドウディスプレイを管理するウインドウディスプレイ管理装置において、実行中タスクに対応する複数のウインドウを表示することができる作業領域を複数提供する複数作業領域提供手段と、前記実行中タスクに対応するアイコンを含むウインドウパネルを表示するウインドウパネル表示手段と、前記ウインドウパネルに表示された前記実行中タスクに対応するアイコンが選択された時に、当該実行中タスクのウインドウに共有アクセスを許可する共有アクセス許可手段とを備えたことを特徴とするウインドウディスプレイ管理装置。

【請求項32】 更に、ウインドウ内にオープンされた各タスク用のアイコンを前記ウインドウパネルに追加するアイコン追加手段を備えたことを特徴とする請求項31

に記載されるウインドウディスプレイ管理装置。

【請求項33】 更に、ユーザからタスクを作動させるためのコマンドを受信する作動コマンド受信手段と、前記タスク用のウインドウを生成するウインドウ生成手段とを備えたことを特徴とする請求項31に記載されるウインドウディスプレイ管理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、コンピュータシステム用のグラフィックユーザインタフェースに関し、更に詳しくは、ウインドウアクセス用パネルの提供方法及びその装置に関するものである。

【0002】

【従来の技術】多くの現代コンピュータシステムは、ユーザグラフィックインタフェースを使用しており、各アクティブタスク用に別々の"ウインドウ"を提供するものである(尚、ここで用いる用語"タスク"は、アプリケーションプログラム及びドキュメントの双方を含むものである)。周知の例として、アップルコンピュータインコーポレーテッドが提供するマッキントッシュユーザインタフェースやマイクロソフトコーポレーション社が提供するウインドウズ95ユーザインタフェースを挙げることができる。このようなユーザインタフェースは、コンピュータをより使い易いものにする。その理由は、これらのユーザインタフェースがユーザに対して、同一のコンピュータ上で同時に実行されるマルチタスクを管理する便利な手法を提供するからである。

【0003】ユーザインタフェースの中には、この概念を複数の作業領域用のサポートを提供することによって更に活用しているものもある。このようなユーザインタフェースの一例としてUnix共有デスクトップ環境(CDE)を挙げることができる。この共有デスクトップ環境は、ヒューレットパッカード社のビジュアルユーザ環境に基づいたものである。共有デスクトップ環境は、サンマイクロシステムズインコーポレーテッド、ヒューレットパッカード社及びIBM社を含む数社が共同して開発したものである。作業領域は、タスクの集合であり、このタスクは、特定のオブジェクトを実行するのに利用されるものである。例えば、ユーザは、個人向け構成用のニュースレター等を作成するために1つの作業領域を有していればよい。ニュースレター用の作業領域は、膨大なタスク用のウインドウを保持するとともに、ワープロ、描画及びデスクトップパブリッシングを含んでいる。同様に、個人向け構成用の作業領域は、カレンダー、スケジューリング及びアドレッシング機能を提供するタスク用のウインドウを保持するものである。複数の作業領域をサポートするインタフェースのユーザは、必要に応じて作業領域を切り換えることができ、コンピュータは、選択された作業領域に属するウインドウを表示する。

【0004】

【発明が解決しようとする課題】しかしながら、複数のウィンドウを保持する機能は、多くの点で優れているとはいえず、多くのウィンドウが大きくなった場合には、邪魔になることがある。多くのウィンドウを保持することは、デスクトップの外観を乱雑なものにし、ユーザがタスクに対応した特定のウィンドウを配置するのを困難にする。ここで使用する用語“タスク”は、コンピュータプロセッサを用いて実行されるソフトウェアを含むものである。ディスプレイスクリーンのサイズと、ユーザによりアクティブに処理されるタスクとに依存して、ウィンドウは、全体が完全に見える状態のものもあれば、ウィンドウの一部のみが見える状態にあるもの、全体が完全に隠蔽されているものもある。

【0005】複数の作業領域をサポートするユーザインタフェースには、次のような不都合もある。例えば、ユーザは、複数の作業領域内に同一のウィンドウを表示したいことがある。この場合、ユーザは、スクリーン上の別位置に配置されたある作業領域内にウィンドウを保持し、又は、このある作業領域内のウィンドウを別の作業領域内の対応するウィンドウとはサイズを変えて保持することになる。

【0006】このような不都合は、次のような事情によるものである。例えば、マイクロソフト社が提供するウィンドウズ95オペレーティングシステムによれば、ユーザスクリーンの下部分に、“タスクバー”が生成されるようになっている。ユーザは、通常、タスクバーをビューの外側に配置し、また、そのスクリーンのエッジ部分にマウスポインタが接触するとタスクバーがスライドしてオープンするように、タスクバーを構築する。ユーザがタスクを作動させる場合、そのタスク用のウィンドウがディスプレイ(デスクトップとも称される)上にオープンされ、そのタスク用のボタンがタスクバーに追加される。ウィンドウズ95のユーザは、対応するウィンドウの右上の角部分にあるボタンを選択することによりタスクを“隠蔽”する選択ができる。これにより、コンピュータは、デスクトップから対応するウィンドウを除去することができるが、そのタスク用のボタンは、タスクバーに残ることになる。ユーザはまた、タスクバーのボタンを指定するのにマウスを用い、マウスボタンをクリックすることにより、隠蔽されたタスクを“見せる”こともできる。これにより、コンピュータは、デスクトップ上に対応するウィンドウを表示することができる。ユーザはまた、タスク用ウィンドウの右上の角部分にあるボタンを選択することにより、タスクを“クローズ”することもできる。これにより、コンピュータは、プログラムの実行を終了させ、デスクトップから対応するウィンドウを除去し、タスクバーからそのタスクのアイコンを除去することができる。

【0007】従って、タスクバーは、相当なウィンドウ処理能力を提供するが、複数の作業領域をサポートする

ユーザインタフェースにおけるウィンドウ管理上の問題を提供するものではない。ウィンドウズ95自身が複数の作業領域をサポートしないのは、この点に原因がある。

【0008】ちなみに、共有デスクトップ環境は、複数の作業領域でウィンドウディスプレイを管理するためのメニュー・ダイアログ機能を提供するが、この機能は、種々の欠点を備えている。例えば、共有デスクトップユーザが特定の作業領域でタスクを作動させると、このタスクに対応するウィンドウは、その作業領域内でのみしかオープンしない。共有デスクトップユーザが他の作業領域内で特定ウィンドウを見ようとする場合、ユーザは、どの作業領域が所望のウィンドウを保持するのか覚えていなければならない。この場合、更に、ユーザは、所望の作業領域に切り換えた後、最初の作業領域を特定するためにメニュー・ダイアログを使用して、その最初の作業領域へ再び切り換える必要がある。更に、ウィンドウは、同一のスクリーン位置に現れると共に、各作業領域内で同一のサイズをとる。即ち、ある作業領域内でのスクリーン位置とサイズにおけるあらゆる変更は、その他の全ての作業領域内のスクリーン位置とサイズとに影響を及ぼす。

【0009】従って、これらの欠点を改善すること、即ち、①多くのウィンドウが大きくなっても邪魔にならず、②複数の作業領域内に同一のウィンドウを簡単に表示でき、③複数の作業領域をサポートするユーザインタフェースにおけるウィンドウ管理上の問題を提供することができ、④タスクに対応するウィンドウを簡単に操作することができるようなシステムを提供することが望まれている。このためには、複数の作業領域をサポートするユーザインタフェース内のウィンドウを、簡単にユーザが選択でき処理することができるシステムを提供する必要がある。

【0010】本発明は、複数の作業領域をサポートするユーザインタフェース内のウィンドウを、簡単にユーザが選択でき処理することができるウィンドウディスプレイ管理方法及びウィンドウディスプレイ管理装置システムを提供することを課題とする。また、本発明は、複数の作業領域をサポートするユーザインタフェース内のウィンドウを、簡単にユーザが選択でき処理することができるウィンドウ管理手順をコンピュータに実行させるための命令を記録したコンピュータ読取可能な記録媒体を提供することを目的とする。

【0011】

【課題を解決するための手段】上記課題を解決するために、本発明に係る方法及びシステムは、コンピュータによって実行可能なタスクに対応するウィンドウディスプレイを管理するものである。まず、請求項1に記載されるウィンドウディスプレイ管理方法は、実行中タスクに対応する複数のウィンドウを表示することができる作業領域を複数提供する複数作業領域提供工程と、前記実行

中タスクに対応するアイコンを含むウインドウパネルを表示するウインドウパネル表示工程と、前記ウインドウパネルに表示された前記実行中タスクに対応するアイコンが選択された時に、当該実行中タスクのウインドウに共有アクセスを許可する共有アクセス許可工程とを含むことを要旨とするものである。

【0012】上記構成を有するウインドウディスプレイ管理方法によれば、複数の作業領域が提供されるが、各作業領域は、実行中タスクに対応する複数のウインドウを表示することができる。次いで、ウインドウパネルが表示されるが、このウインドウパネルは、その実行中タスクに対応するアイコンを含む。そして、このアイコンが選択された時に、その実行中タスクのウインドウに共有アクセスが許可されることになる。

【0013】この場合に、ウインドウ内にオープンされた各タスク用のアイコンを前記ウインドウパネルに追加するアイコン追加工程を含むようにするとよい。

【0014】更に、ユーザからタスクを作動させるためのコマンドを受信する作動コマンド受信工程と、前記タスク用のウインドウを生成するウインドウ生成工程とを含むようにするとよい。

【0015】更に、ユーザから被選択作業領域内のウインドウを隠蔽するためのコマンドを受信する隠蔽コマンド受信工程と、前記被選択作業領域からのみ前記ウインドウを除去するウインドウ除去工程と、前記ウインドウパネル内に前記ウインドウを表すアイコンを維持するアイコン維持工程とを含むようにするとよい。この場合、更に、前記被選択作業領域内の前記ウインドウのサイズと位置とを記録する記録工程を含むようにするとよい。

【0016】更に、特定の作業領域内で、ユーザから前記ウインドウパネル内のアイコンを選択するためのコマンドを受信する選択コマンド受信工程と、その被選択アイコンに対応するウインドウを前記作業領域の前面に表示する前面ウインドウ表示工程とを含むようにするとよい。

【0017】更に、ユーザからウインドウを終了するためのコマンドを受信する終了コマンド受信工程と、前記ウインドウを当該ウインドウが表示されている各作業領域から除去する非限定ウインドウ除去工程と、その終了ウインドウに対応するアイコンを前記ウインドウパネルから削除するアイコン削除工程とを含むようにするとよい。

【0018】更に、複数の作業領域内にウインドウを表示するウインドウ表示工程と、他の作業領域内のウインドウのサイズに影響を与えることなく、ある作業領域内のウインドウのサイズを変更するサイズ変更工程とを含むようにしてもよい。

【0019】更に、複数の作業領域内にウインドウを表示するウインドウ表示工程と、他の作業領域内のウインドウの位置に影響を与えることなく、ある作業領域内の

ウインドウの配置を変更する配置変更工程とを含むようにしてもよい。

【0020】更に、作業領域切換スイッチの後ろに前記ウインドウパネルを隠蔽する隠蔽工程と、ユーザから前記作業領域切換スイッチを選択するためのコマンドを受信する選択コマンド受信工程と、前記スイッチを選択するユーザコマンドにตอบสนองして前記ウインドウパネルを表示するウインドウパネル表示工程とを含むようにしてもよい。

【0021】次に、請求項11に記載される本発明に係るコンピュータ読取可能な記録媒体は、実行中タスクに対応する複数のウインドウを表示することができる作業領域を複数提供する複数作業領域提供手順と、前記実行中タスクに対応するアイコンを含むウインドウパネルを表示するウインドウパネル表示手順と、前記ウインドウパネルに表示された前記実行中タスクに対応するアイコンが選択された時に、当該実行中タスクのウインドウに共有アクセスを許可する共有アクセス許可手順とを含むウインドウディスプレイ管理手順をコンピュータに実行させるための命令を記録したものである。

【0022】上記構成を有するコンピュータ読取可能な記録媒体によれば、これをコンピュータに実装することにより、上記ウインドウディスプレイ管理手順が実行されることになる。

【0023】次に、請求項21に記載されるコンピュータによって実行可能なタスクに対応するウインドウディスプレイを管理するウインドウディスプレイ管理装置は、プログラム命令を保持したメモリと、前記プログラム命令を使用するように構成されたプロセッサとを備え、当該プロセッサが、実行中タスクに対応する複数のウインドウを表示することができる作業領域を複数提供する複数作業領域提供手段と、前記実行中タスクに対応するアイコンを含むウインドウパネルを表示するウインドウパネル表示手段と、前記ウインドウパネルに表示された前記実行中タスクに対応するアイコンが選択された時に、当該実行中タスクのウインドウに共有アクセスを許可する共有アクセス許可手段とを備えたことを要旨とするものである。

【0024】上記構成を有するウインドウディスプレイ管理装置によれば、複数の作業領域が提供されるが、各作業領域は、実行中タスクに対応する複数のウインドウを表示することができる。次いで、ウインドウパネルが表示されるが、このウインドウパネルは、その実行中タスクに対応するアイコンを含む。そして、このアイコンが選択された時に、その実行中タスクのウインドウに共有アクセスが許可されることになる。

【0025】

【発明の実施の形態】以下に本発明に係る一実施形態について、添付した図面を参照して詳細に説明する。尚、全図を通して同一又は類似の部材を参照するにあたって



は、できる限り同一の符号を用いている。

#### 【0026】A. 概略

本発明に係る装置及び方法は、複数の作業領域をサポートするグラフィックユーザインタフェースを用いて操作するためのものである。本発明に係る装置及び方法は、ウィンドウがどの作業領域に存在しているかに拘わらず、ウィンドウ内にオープンされた各タスクを表すアイコンを保持するスライド可能なウィンドウパネルを用いたものである。新たなタスクがウィンドウ内にオープンされる場合には、そのタスクを表すアイコンがそのウィンドウパネルに追加されることになる。

【0027】ユーザは、ある作業領域内のウィンドウを隠蔽することもできるが、そのウィンドウは、そのウィンドウが存在していた他のいずれかの作業領域内で出すことができる。コンピュータは、そのウィンドウパネル内で隠蔽されたウィンドウを表すアイコンを維持している。これにより、そのウィンドウは、その後に出すことができるようになっていく。更に、コンピュータは、ウィンドウの位置とサイズとを記憶している。従って、その後ウィンドウを出す場合には、ウィンドウは、隠蔽された時に存在していたのと同様に表示されることになる。一方、ユーザは、もはや必要でなくなったウィンドウをクローズすることもできる。そのウィンドウは、それが存在していた全ての作業領域から除去され、対応するアイコンがウィンドウパネルから除去されることになる。

【0028】ユーザは、ウィンドウパネルからアイコンを選択することによりウィンドウを出すこともできる。ウィンドウが作業領域内に既にオープンされていても、その前面が他のウィンドウによって遮られている場合には、ウィンドウは、ディスプレイの前面へ出すことになる。ウィンドウが隠蔽されている場合には、それが隠蔽される前に存在していた作業領域内の同一の位置及びサイズに記憶しなおされる。ウィンドウを作業領域内で一度も出していない場合には、ウィンドウは、デフォルトの位置にデフォルトのサイズで出すことになる。ユーザはまた、他の作業領域内の対応するウィンドウのサイズ及び位置に影響を与えることなく、作業領域内のウィンドウの位置及びサイズを変更することができる。

#### 【0029】B. アーキテクチャ

図1は、本発明に係る一実施形態が実装されるコンピュータシステム100のブロック構成を示したものである。システム100は、コンピュータ110を含み、このコンピュータ110は、ネットワーク170を介してサーバ180に接続されている。ネットワーク170としては、ローカルエリアネットワーク(LAN)、ワイドエリアネットワーク(WAN)又はインターネットを適用することができる。本実施例においては、コンピュータ110として、ネットワークコンピュータが採用される。システム100は、HotJava™ Views™ ユーザ環境で使用するのに適しているが、いわゆる

当業者に自明であるように、本発明に係る方法及び装置は、その他の好適なユーザ環境にも適用し得るものである。HotJava™ Views™ は、サンマイクロシステムズインコーポレーテッドにより開発されたネットワークコンピュータ用のグラフィックユーザインタフェースである。この詳細については、サンマイクロシステムズインコーポレーテッド:1998年版”ネットワークコンピュータ用のThe HotJava Views™ ユーザ環境の設計”に説明されているので、そちらを参照されたい。尚、このドキュメントは、1998年5月27日付けで、インターネットのホームページ<http://java.sun.com/products/hotjavaviews/hjv.white.html>に掲載されている。HotJava Viewsは、Java™ プログラミング言語により記述されており、また、この言語を介したユーザインタフェースを処理するものである。Javaプログラミング言語については、例えば、アジソンウェスレイ社:1996年版、ジェイムズ・ゴスリング、ビル・ジョイ、ガイ・スティール共著の”The Java™ 言語仕様”に記載されているのでこちらを参照されたい。Sun, Sun Microsystems, JavaSoft, the Sun Logo, Java及びHotJava Viewsは、米国及びその他海外のサンマイクロシステムズインコーポレーテッドの商標又は登録商標である。

【0030】ネットワークコンピュータ110は、種々のコンポーネントを備えており、これらのコンポーネントは、全てシステムバス120を介して内部的に接続されている。バス120は、例えば、ネットワークコンピュータ110のコンポーネントを接続する双方向システムバスである。このバス120は、メモリ125をアドレッシングするために32本のアドレス線を保持し、それらのコンポーネント間でデータ転送を行うために32ビットのデータバスを保持している。別々のデータ/アドレス線に代えて、マルチプレクサのデータ/アドレス線を用いてもよい。

【0031】ネットワークコンピュータ110は、プロセッサ115を備えており、このプロセッサ115は、メモリ125に接続されている。プロセッサ115としては、680X0プロセッサ等のモトローラ社により製造されたマイクロプロセッサや、80X86、Pentiumプロセッサ等のインテル社により製造されたプロセッサの他、サンマイクロシステムズインコーポレーテッドにより製造されたSPARC™ マイクロプロセッサが適用し得るものである。しかしながら、これに限定されるものではなく、その他の好適なマイクロプロセッサの他、マイクロコンピュータ、ミニコンピュータ又はメインフレームコンピュータもまた適用し得るものである。メモリ125は、ランダムアクセスメモリ(RAM)、リードオンリメモリ(ROM)、ビデオメモリ及び大容量記憶装置を備えていてもよい。大容量記憶装置は、固定メディア及び脱着可能メディアの両者を含むものであってもよい(即ち、磁気記憶、光学記憶又は磁気光学記憶システムその他の有効な大容量記憶技術が適用し得るものである)。

【0032】ユーザは、通常、キーボード130やポインティングデバイス(マウス135等)を介してネットワークコンピュータ110へ情報を入力する。情報の入力には、その他の入力デバイスを用いてもよい。そして情報入力に対して、ディスプレイスクリーン140を介して情報がユーザへ送られることになる。

【0033】コンピュータ110は、ネットワークインタフェース145(例えば、イーサネットやダイヤルアップ電話回線等)を介してネットワーク170上で他のコンピュータと情報通信を行うものである。従って、コンピュータ110は、ネットワーク170を介して、メッセージを送信すると共にデータを受信する。尚、メッセージやデータには、プログラムコードが含まれる。本発明によれば、このようなダウンロードされたタスクは、本発明の一実施形態に係る、ユーザ環境用アプリケーションプログラムである。受信されたコードは、受信と同時に、プロセッサ115に実行させてもよいし、その後実行させるためにメモリ125に記憶するようにしてもよい。尚、アプリケーションコードは、あらゆる形態の他のコンピュータプログラム媒体で処理することができる。ここで説明されるコンピュータシステムは、本発明の一実施形態として示したものである。従って、本発明は、あらゆるタイプのコンピュータシステムその他、あらゆるタイプのプログラミング又はプロセッシング環境で実装可能である。

【0034】ネットワークコンピュータ110によって要求されるオペレーティングシステム、タスク及びデータは、サーバ180に集中的に記憶され、必要に応じてネットワーク170を介してロードされるものである。まず、ネットワークコンピュータ110に電源が投入されると、サーバ180は、コンピュータ110に、オペレーティングシステム(例えば、サンマイクロシステムズインコーポレーテッドにより開発されたJavaOS 1.0)と共にブートイメージを提供する。サーバ180はまた、ネットワークコンピュータ110に、HotJava Viewsユーザインタフェースを提供するものである。このユーザインタフェースは、複数の作業領域、例えば、電子メール作業領域、電子カレンダー作業領域、組織内人員配置ディレクトリを保持する作業領域、及び、イントラネット又はインターネット上のドキュメント閲覧用のウェブブラウザ閲覧作業領域をサポートするものである。実際にインストールされたタスク集合及びこれに対応する有効な作業領域は、システム管理者がどのようにそのシステムを構築するかに応じて決まるものである。

【0035】図2は、本発明に係る作業領域を示すユーザインタフェーススクリーンである。同図において、セレクトタ210は、各作業領域に対応するアイコンを保持するスクリーンの左側縦長欄である。このセレクトタ210は、作業領域切替スイッチとして機能し、これを用いてユーザは、異なる作業領域間を行ったり来たりすることができる。ユーザは、セレクトタ210内のアイコンのどれ

かをクリックすることにより、作業領域を選択することができる。これに応答して、コンピュータ110は、その作業領域に保持されるウィンドウを表示する。

【0036】セレクトタ210は、HotJava Viewsの一部であるユーザ環境アプリケーションプログラムにより生成されるものである。このセレクトタ210は、スタートアップ中にサーバ180からネットワークコンピュータ110に提供されるものである。ユーザ環境アプリケーションは、HotJava Viewsで使用されるグラフィックユーザインタフェースを生成し管理する元となるものである。例えば、ユーザ環境アプリケーションプログラムは、内部データ構造をメモリ125内に保持する。この内部データ構造には、どのウィンドウがどの作業領域にオープンされたかが当該ウィンドウのその時点での位置とサイズとに従って記録されている。

【0037】図2に示したように、セレクトタ210は、複数の作業領域(メール、個人向け構成及び作業領域N等のその他の作業領域)に対応するアイコンを保持している。個人向け構成用の作業領域は、アクティブになっており、これは、セレクトタ210内のアイコンの周りの太線によって表されている。個人向け構成用作業領域は、2つのウィンドウを保持している。1つは、ノートパッドタスク(これにより、ユーザは短いメッセージを生成し記憶することができる。)に対応するものであり、もう1つは、スケジュールタスク(これは、特定の日、週、月の活動を示すものである。)に対応するものである。

【0038】ユーザインタフェースはまた、スライド可能なウィンドウパネル220を保持しており、このウィンドウパネル220は、見える状態か隠蔽状態かに関わらず、各作業領域内の各ウィンドウ用のアイコンを保持するものである。デフォルトとして、このアイコンリストは、アルファベット順に上から並べたものである。しかしながら、ユーザは、必要であればアイコンリストの順番を好きなように変えることができる。図2に示したように、ウィンドウパネル220は、3つのウィンドウ用のアイコンを保持している。即ち、インボックス、ノートパッド及びスケジュールである。たとえ、被選択作業領域に2つのウィンドウしか表示されていない場合でも、ウィンドウパネル220は、3つのウィンドウに対応するアイコンを保持している。

【0039】本実施形態においては、ウィンドウパネル220は、セレクトタ210に隣接して配置されている。ユーザインタフェースの乱雑さを最小限におさえるために、ウィンドウパネル220は、通常、隠蔽されている。隠蔽状態にある場合、その存在は、スクリーン140の左端に沿って、細い長方形によってユーザに示される。ユーザは、スクリーン140の左端にポインタを移動させることにより、ウィンドウパネル220をオープンさせることができる。これにより、ウィンドウパネル220は、セレクトタ210の下から図2に示した位置へスライドして出てく

ることになる。

【0040】図3は、本発明の一実施形態における他の作業領域を示したユーザインタフェーススクリーンである。図3に示したスクリーンは、図2に示したスクリーンに対応している。図3においては、メール用作業領域は、アクティブになっており、これは、セクタ210内のアイコンの周りの太線によって表されている。また、スクリーンエリア225内に表示されている。メール用作業領域は、2つのウィンドウを保持している。1つは、インボックスタスク(これは、届いた電子メールのメッセージを示す。)に対応するものであり、もう1つは、図2に示したノートパッドタスクに対応するものである。また、たとえ、被選択作業領域に2つのウィンドウしか表示されていない場合でも、ウィンドウパネル220は、3つのウィンドウに対応するアイコンを保持している。

【0041】図3は、また、描画ソフト230を示す。この描画ソフト230は、作動可能なタスクに対応するアイコンを保持しており、ノートパッド、スプレッドシート及びタスクX等のその他のタスクに対応するアイコンを保持している。

#### 【0042】C. アーキテクチャの動作

図2及び図3に示したスクリーンディスプレイに基づいて、本発明の一実施形態について更に説明する。いわゆる当業者に自明であるように、ここに開示した組み合わせは、本発明を実現するための典型例である。

【0043】図4は、本発明に係る方法の一実施形態に係り、図3に示したノートパッドタスクを作動させるために実行される工程を示したフローチャートである。この処理は、ユーザがセクタ210内のアイコンをクリックすることにより、ユーザからメール用作業領域を選択するためのコマンドをシステム100が受信した時から開始する(ステップ310)。メール用作業領域が選択された後、システム100は、ユーザからノートパッドタスク作動用のコマンドを受信する(ステップ320)。本実施例においては、ユーザは、これをマウス135を用いてポインタをスクリーンの下部へ移動させることにより行う。これにより、描画ソフト230がオープンする。この描画ソフト230は、作動可能な種々のタスク用のアイコンを保持している。ユーザは、ノートパッドに対応するアイコンを指定し、マウスでボタンをクリックして該当するタスクを作動させる。

【0044】ユーザからのノートパッドタスクを作動させるためのコマンドにตอบสนองして、プロセッサ115は、ノートパッドタスクが既にメール用作業領域内のウィンドウに表示されているか否かを決定するためにチェックを行う(ステップ330)。表示されている場合(本実施例においては)、プロセッサ115は、対応するウィンドウをディスプレイの前面へもってくるようにし(ステップ360)、これにより、該当する処理が終わる。ノートパッドタスクが未だメール用作業領域内のウィンドウに表示さ

れていない場合には、プロセッサ115は、ノートパッドタスクを保持するウィンドウを生成して、ディスプレイスクリーン140に表示する(ステップ340)。更に、プロセッサ115は、作動したばかりのノートパッドタスクに対応するアイコンをウィンドウパネル220に追加する(ステップ350)。

【0045】図5は、本発明の一実施形態に係り、メール用作業領域内のノートパッドウィンドウを隠蔽するために実行される工程を示したフローチャートである。この図5に示したフローチャートは、ノートパッドウィンドウがメール用作業領域及び個人向け構成用作業領域の両方に表示されていると仮定した場合に該当するものである。この処理は、ユーザがセクタ210内のアイコンをクリックすることにより、ユーザからメール用作業領域を選択するためのコマンドをシステム100が受信した時から開始する(ステップ410)。これにตอบสนองして、プロセッサ115は、スクリーン140上に作業領域内に示された全てのウィンドウを表示する。その後、システム100は、ユーザからノートパッドウィンドウを隠蔽するためのコマンドを受信する(ステップ420)。これは、ノートパッドウィンドウの枠内の指定されたボタンをユーザがクリックすることによりなされるものである。このコマンドにตอบสนองして、プロセッサ115は、ノートパッドウィンドウのサイズと位置とをメモリ125に記憶する(ステップ430)。これにより、ユーザがその後に、そのウィンドウを出す場合、そのウィンドウは、隠蔽前に作業領域内に存在していたのと同じ位置に同一のサイズで表示されることになる。プロセッサ115は、最後に、ノートパッドウィンドウをメール用作業領域のディスプレイから除去する(ステップ440)。

【0046】尚、ウィンドウは、メール用作業領域内に隠蔽されるが、他の作業領域(例えば、図2に示した個人向け作業領域)にオープンされている対応するウィンドウバージョンのいずれかは、その作業領域内でそのまま見える状態で維持されている。更に、ノートパッドウィンドウは、カレント作業領域内に隠蔽されるが、アイコンは、そのウィンドウがウィンドウパネル220内で維持されていることを示す。これにより、ユーザは、その後もウィンドウパネル220を用いることにより、メール用作業領域内にノートパッドウィンドウを出すことができる。

【0047】図6は、本発明の一実施形態に係り、隠蔽されたウィンドウ(全部又は一部が隠蔽されたウィンドウ)を出すために実行される工程を示したフローチャートである。この図6に示したフローチャートは、ノートパッドウィンドウがメール用作業領域及び個人向け構成用作業領域の両方で隠蔽されていると仮定した場合に該当するものである。この処理は、ユーザがセクタ210内のアイコンをクリックすることにより、ユーザから作業領域を選択するためのコマンドをシステム100が受信

した時から開始する(ステップ505)。例えば、ユーザによりメール用作業領域が選択される。システム100は、その後、ユーザからウインドウパネル220にアクセスするコマンドを受信する(ステップ510)。これは、ユーザがスクリーン140の左端を指定することによりなされるものである。ユーザの指定にตอบสนองして、ウインドウパネル220は、スライドしてセレクト210に隣接してオープンする(ステップ515)。

【0048】ウインドウパネル220が表示されると、システム100は、ユーザからウインドウを表示するためのコマンドを受信する。このコマンドは、ユーザがノートパッドウインドウを表すアイコンをマウス135を用いてクリックすることにより、送出される(ステップ520)。これにตอบสนองして、プロセッサ115は、ノートパッドウインドウが既にカレント作業領域内に表示されているか否かを決定するためにチェックを行う(ステップ525)。既に表示されている場合には、プロセッサ115は、ノートパッドウインドウをディスプレイ140の前面へ移動させる(ステップ530)。これにより、ユーザは、カレント作業領域に既に存在するウインドウを前面へもってこることができるが、その他のウインドウによって、その全部又は一部が隠蔽されたままとなる。

【0049】被選択ウインドウが未だカレント作業領域に表示されていない場合には(本実施例において)、プロセッサ115は、被選択ウインドウが前にカレント作業領域に表示されていたか否かを決定するためにチェックを行う(ステップ535)。前に表示されていた場合には、プロセッサ115は、前に表示されていたウインドウのサイズと位置とを、メモリ125から再度呼び出す(ステップ540)。前に表示されなかった場合には、デフォルトのサイズと位置とがウインドウ用に用いられる。次いで、プロセッサ115は、記憶した又はデフォルトのパラメータのいずれかを用いて、ノートパッドウインドウをメール用作業領域に表示する(ステップ545)。尚、ウインドウは、メール用作業領域に表示されるが、個人向け構成用作業領域等のその他の作業領域内の対応するウインドウの表示(又は非表示)に影響を与えることはない。

【0050】図7は、本発明の一実施形態に係り、個人向け構成用作業領域からノートパッドウインドウを終了させるために実行される工程を示したフローチャートである。この図7に示したフローチャートは、ノートパッドウインドウがメール用作業領域及び個人向け構成用作業領域の両方で表示されていると仮定した場合に該当するものである。その処理は、ユーザから個人向け作業領域を選択するためのコマンドをシステム100が受信した時から開始する(ステップ610)。個人向け構成用作業領域がウインドウを保持していない場合には、セレクト210を用いて適当な作業領域が選択されることになる。システム100は、その後、ユーザからノートパッドウインドウを終了させるためのコマンドを受信する(ステップ6

20)。コマンドは、ユーザがノートパッドウインドウ上の指定されたボタンをクリックすることにより、送出されるものである。あるいは、ユーザがウインドウパネル220から直接ウインドウを終了させてもよい。この場合、ユーザは、該当するウインドウを表すアイコンを指定し、“終了”コマンドを保持するポップアップメニューを用いればよい。これにตอบสนองして、プロセッサ115は、ノートパッドウインドウを全ての作業領域の表示から除去する(ステップ630)。プロセッサ115はまた、ノートパッドウインドウを表すアイコンをウインドウパネル220から除去し(ステップ640)、これにより、処理が終わる。

【0051】図8は、本発明の一実施形態に係り、メール用作業領域内のノートパッドウインドウの位置又はサイズを変更するために実行される工程を示したフローチャートである。この図8に示したフローチャートは、ノートパッドウインドウがメール用作業領域及び個人向け構成用作業領域の両方で表示されていると仮定した場合に該当するものである。この処理は、セレクト210を用いるユーザからメール用作業領域を選択するためのコマンドをシステム100が受信した時から開始する(ステップ710)。これにตอบสนองして、プロセッサ115は、作業領域内に保持される全てのウインドウをスクリーン140上に表示する。システム100は、その後、ノートパッドウインドウを選択しているユーザから位置又はサイズの変更のためのコマンドを受信する(ステップ720)。これは、マウス135を用いてなされる。システム100は、その後、ユーザからノートパッドウインドウの位置又はサイズを変更するためのコマンドを受信する(ステップ730)。これもまた、マウス135を用いてなされる。ユーザの指令にตอบสนองして、プロセッサ115は、ノートパッドウインドウのサイズ又は位置をディスプレイスクリーン140上で変更し(ステップ740)、これにより処理が終わる。尚、メール用作業領域内のノートパッドウインドウの位置又はサイズの変更を行うためのユーザの指令は、個人向け作業領域等、他の作業領域内の対応するノートパッドウインドウの位置又はサイズに影響を与えることはない。

#### 【0052】D. 結論

以上説明したように、本発明に係る方法及び装置によれば、ユーザは、複数の作業領域をサポートするユーザインタフェース内のウインドウを簡単に選択でき処理することができる。以上説明した本発明の一実施形態は、例示及び説明を目的としたものである。例えば、上述した実施形態は、ソフトウェアを含むものであるが、本発明の実施形態は、これに限定されるものではなく、ハードウェアとソフトウェアとの組み合わせ、又は、ハードウェアのみによっても、本発明の要旨は実現され得るものである。上述した記載から導き出される事項及び本発明の実施上必要とされる事項から、図4乃至図7に関して説明された工程に対し、種々の変容を加えることができ

る。

【0053】本発明に係るシステム及び方法は、分散型システムやJava™プログラミング環境において動作するものとして説明してきたが、いわゆる当業者にとって自明であるように、本発明は、その他のシステムやプログラミング環境においても実現され得るものである。更に、本発明の要旨とするところは、メモリに記憶されているものとして説明したが、いわゆる当業者にとって自明であるように、その他のタイプのコンピュータ読取可能な媒体に記憶することもできる。そのようなコンピュータ読取可能な記録媒体としては、例えば、ハードディスク、フロッピーディスク又はCD-ROM等の補助記憶デバイス、インターネットからのキャリアウェイブ、又はランダムアクセスメモリ(RAM)やリードオンリメモリ(ROM)等のその他の形態のものが好適なものとして挙げられる。本発明の範囲は、特許請求の範囲及びこれを裏付ける記載によって定まるものである。

#### 【0054】

【発明の効果】本発明に係る請求項1乃至10に記載されるウィンドウディスプレイ管理方法によれば、実行中タスクに対応する複数のウィンドウを表示することができる複数の作業領域が提供され、当該実行中タスクに対応するアイコンを含むウィンドウパネルが表示される。このウィンドウパネルは、前記複数のウィンドウ内に各タスクがオープンされると当該タスク用のアイコンが追加されるようになっている。従って、前記作業領域内に前記ウィンドウが存在しているか否かに関わらず、このアイコンを選択することにより、前記実行中タスクのウィンドウに共有アクセスが許可されることになる。従って、ユーザは、複数の作業領域をサポートするユーザインタフェース内のウィンドウを簡単に選択でき処理することができる。

#### 【図面の簡単な説明】

【図1】本発明に係る一実施形態が実装されるコンピュ

ータシステムのブロック図である。

【図2】本発明の一実施形態における作業領域を示したユーザインタフェーススクリーンを示した図である。

【図3】本発明の一実施形態における他の作業領域を示したユーザインタフェーススクリーンを示した図である。

【図4】本発明の一実施形態に係り、タスクを起動させるために実行される工程を示したフローチャートである。

【図5】本発明の一実施形態に係り、ウィンドウを隠蔽するために実行される工程を示したフローチャートである。

【図6】本発明の一実施形態に係り、隠蔽したウィンドウを示すために実行される工程を示したフローチャートである。

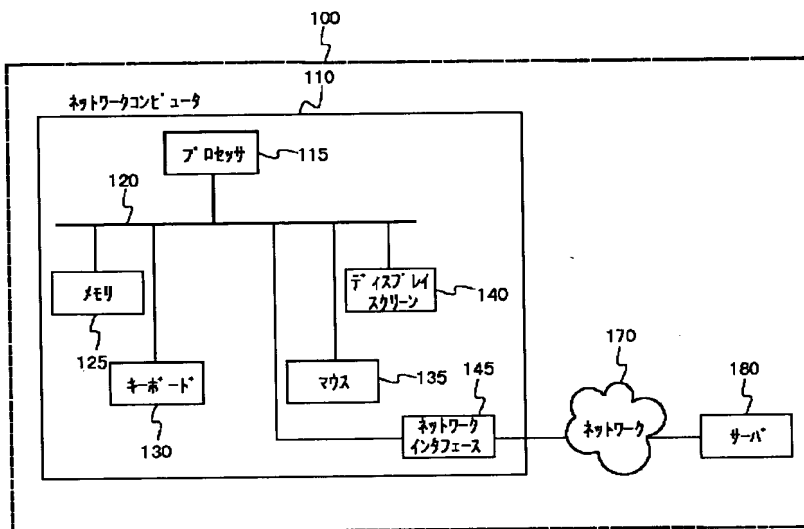
【図7】本発明の一実施形態に係り、ウィンドウを終了させるために実行される工程を示したフローチャートである。

【図8】本発明の一実施形態に係り、ウィンドウの位置又はサイズを変更するために実行される工程を示したフローチャートである。

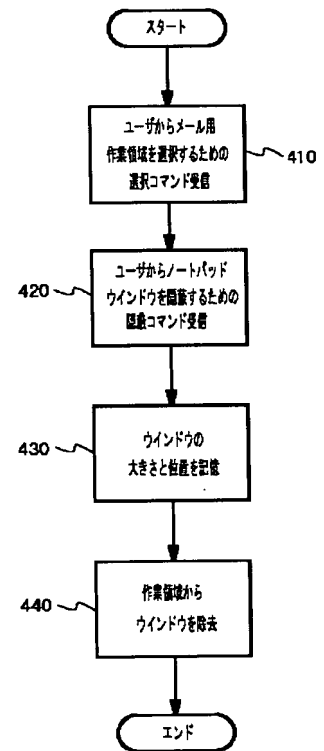
#### 【符号の説明】

- 100 コンピュータシステム
- 110 ネットワークコンピュータ
- 115 プロセッサ
- 120 システムバス
- 125 メモリ
- 130 キーボード
- 135 マウス
- 140 ディスプレイスクリーン
- 145 ネットワークインタフェース
- 170 ネットワーク
- 180 サーバ

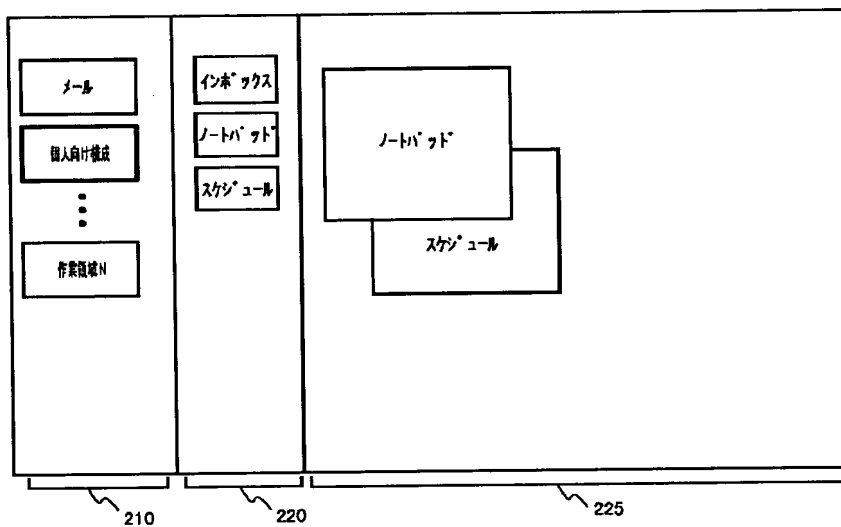
【図1】



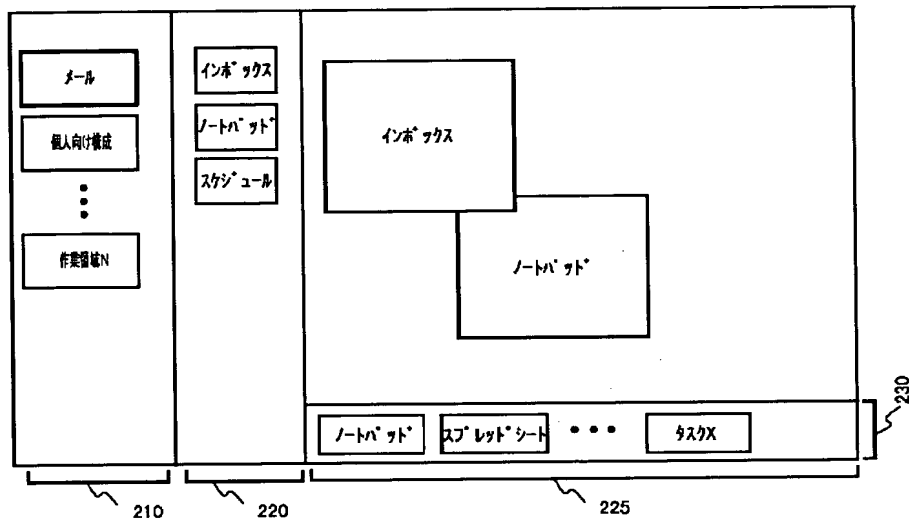
【図5】



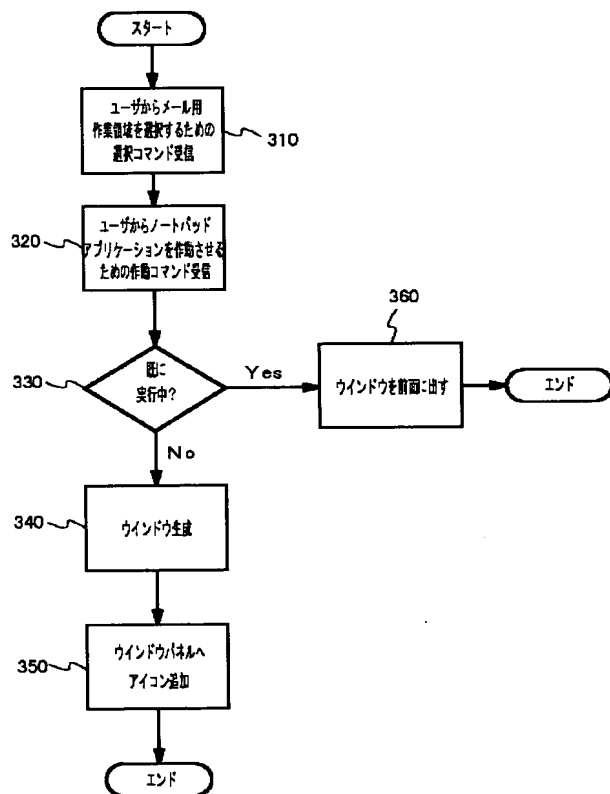
【図2】



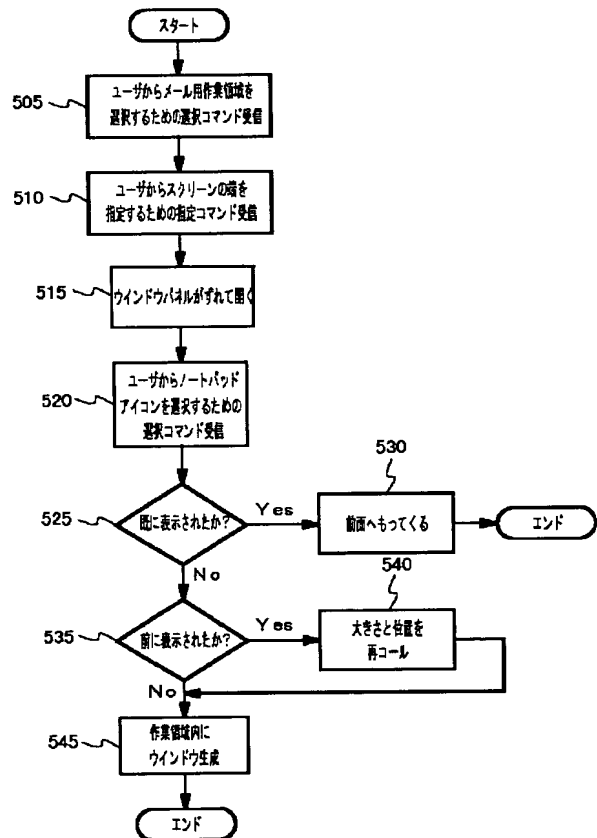
【図3】



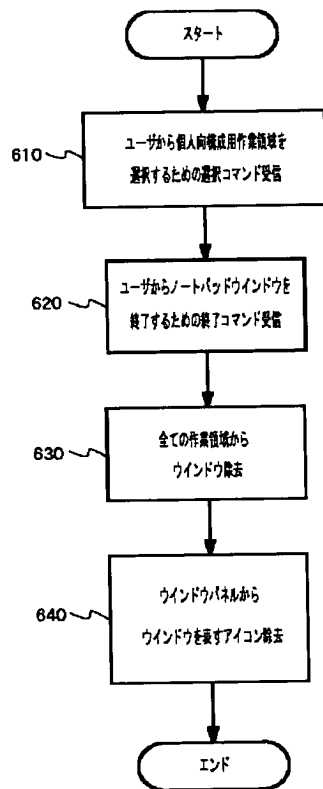
【図4】



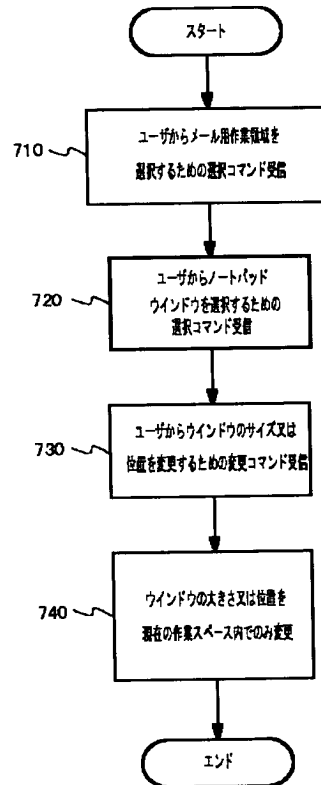
【図6】



【図7】



【図8】



フロントページの続き

(72)発明者 ジョージ サラカン  
 アメリカ合衆国 94087 カリフォルニア  
 州、サニーベイル、シェイエン ドライブ  
 544



## 【 外 国 語 明 細 書 】

## METHODS AND APPARATUS FOR A WINDOW ACCESS PANEL

BACKGROUND OF THE INVENTIONA. Field of the Invention

This invention relates generally to graphic user interfaces for computer systems and, more particularly, to methods and apparatus for providing a window access panel.

5 B. Description of the Related Art

Many modern computer systems employ graphic user interfaces that provide a separate "window" for each active task (as used herein, the term "task" includes both application programs and documents). Familiar examples include the Macintosh user interface from Apple Computer, Inc., and the Windows 95 user interface from Microsoft Corp. Such user interfaces facilitate computing because they provide a convenient way  
10 for a user to manage multiple tasks that are concurrently executing on the same computer.

Some user interfaces take this concept a step further by providing support for multiple workspaces. An example of such a user interface is the Unix Common Desktop Environment (CDE), which is based upon the Hewlett-Packard Visual User Environment.  
15 CDE was developed jointly by a group of companies including Sun Microsystems, Inc., Hewlett-Packard, and International Business Machines. A workspace is a collection of tasks that are employed to accomplish a specific objective. For example, a user may have one workspace for creating a newsletter and another for personal organization. The newsletter workspace may contain windows for numerous tasks, including word  
20 processing, drawing, and desktop publishing. Similarly, the personal organization workspace may contain windows for tasks that provide calendaring, scheduling, and addressing capabilities. A user of an interface that supports multiple workspaces can

switch between those workspaces as needed, and the computer displays windows belonging to the selected workspace.

5        This capability of having multiple windows, while elegant in many respects, becomes somewhat cumbersome when the number of windows becomes large. Having numerous windows leads to a cluttered desktop appearance and makes it difficult for the user to locate a specific window corresponding to a task. As used herein, the term “task” may include software executing using a computer processor. Depending on the size of the display screen and on the task actively manipulated by the user, some of the windows may be completely visible while others may be partially visible or completely hidden.

10        Additional problems arise in user interfaces that support multiple workspaces. For example, a user may desire to display the same window in more than one workspace. In doing so, the user may desire to have a window in one workspace located at a different screen position, or sized differently than, the corresponding window in another workspace.

15        These problems have been addressed somewhat by others. For example, Microsoft has created a “Taskbar” for the Windows 95 operating system that typically resides at the bottom of the user’s screen. A user may configure the Taskbar to reside out of view, normally, and to slide open when the mouse pointer touches the edge of the screen. Whenever a user launches a task, a window for that task is opened on the display  
20        (also referred to as a desktop), and a button for that task is added to the Taskbar. A Windows 95 user can choose to “hide” a task by selecting a button in the top right corner of the corresponding window. This causes the computer to remove the corresponding window from the desktop, but the button for the task remains in the Taskbar. The user can also “show” a previously hidden task by using a mouse to point at a button in the

Taskbar and clicking the mouse button. This causes the computer to display the corresponding window on the desktop. The user can also "close" a task by selecting a button in the top right corner of the task's window. This causes the computer to terminate execution of the program, remove the corresponding window from the desktop, and remove the task's icon from the Taskbar.

Although the Taskbar provides significant window manipulation capabilities, it does not address the problem of window management in a user interface that supports multiple workspaces. This is primarily because Windows 95, itself, does not support multiple workspaces.

CDE provides a menu and dialog mechanism for managing window displays in multiple workspaces, but this mechanism has several deficiencies. For example, when a CDE user launches a task in a particular workspace, a corresponding window is opened in that workspace only. If the CDE user desires to view that particular window in another workspace, the user must remember which workspace contains the desired window, switch to that workspace, use a menu and dialog to specify the first workspace, and then switch back to the first workspace. Moreover, the window appears in the same screen position and at the same size in each workspace; any change in screen position or size in one workspace affects the screen position and size in all other workspaces. There is therefore a need for a system that alleviates these problems and allows a user to easily select and manipulate windows in a user interface that supports multiple workspaces.

#### SUMMARY OF THE INVENTION

Systems and methods consistent with the present invention, as embodied and broadly described herein, manage the display of windows corresponding to tasks executable by a computer. Multiple workspaces are provided, each of which is capable

of displaying multiple windows corresponding to executing tasks. A window panel may be displayed that includes icons corresponding to the executing tasks, and permits shared access to a window of the executing tasks upon selection of a corresponding icon displayed in the window panel.

5           In accordance with the invention, a computer-readable medium contains instructions for managing the display of windows corresponding to tasks executable by a computer. This is accomplished by providing multiple workspaces, each of which is capable of displaying multiple windows corresponding to executing tasks. A window panel may be displayed that includes icons corresponding to the executing tasks, and  
10           permits shared access to a window of the executing tasks upon selection of a corresponding icon displayed in the window panel.

          In accordance with the invention, a system for managing windows in a user interface that supports multiple workspaces comprises a memory having program instructions, and a processor. The processor is configured to use the program instructions  
15           to providing multiple workspaces, each of which is capable of displaying multiple windows corresponding to executing tasks. The processor is also configured to display a window panel that includes icons corresponding to the executing tasks, and to permit shared access to a window of the executing tasks upon selection of a corresponding icon displayed in the window panel.

20                           BRIEF DESCRIPTION OF THE DRAWINGS

          The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate an embodiment of the invention and, together with the description, serve to explain the advantages and principles of the invention. In the

drawings,

FIG. 1 is a block diagram of a computer system in which systems consistent with the present invention may be implemented;

5 FIG. 2a is a representative user interface screen showing a workspace consistent with the present invention;

FIG. 2b is a representative user interface screen showing another workspace consistent with the present invention;

FIG. 3 is a flow diagram of operations performed to launch a task consistent with the present invention;

10 FIG. 4 is a flow diagram of operations performed to hide a window consistent with the present invention;

FIG. 5 is a flow diagram of operations performed to show a hidden window consistent with the present invention;

15 FIG. 6 is a flow diagram of operations performed to close a window consistent with the present invention; and

FIG. 7 is a flow diagram of operations performed to reposition or resize a window consistent with the present invention.

#### DETAILED DESCRIPTION

20 Reference will now be made in detail to an implementation of the present invention as illustrated in the accompanying drawings. Wherever possible, the same reference numbers will be used throughout the drawings and the following description to refer to the same or like parts.

#### A. Overview

Systems and methods consistent with the present invention operate in a graphic user interface that supports multiple workspaces. The systems and methods employ a sliding window panel that contains icons representing every task opened into a window, regardless of the workspace in which it exists. Whenever a new task is opened into a window, an icon representing that task is added to the window panel.

A user may hide a window in one workspace, but that window remains visible in any other workspaces in which it existed. The computer retains an icon representing the hidden window in the window panel so that the window may be subsequently shown. Moreover, the computer stores the location and size of the window so that when it is subsequently shown, it may be displayed as it existed when it was hidden. Alternatively, a user may close a window that is no longer needed. The window is removed from every workspace in which it existed, and the corresponding icon is removed from the window panel.

A user may show a window by selecting its icon from the window panel. If that window is already open in the workspace but obscured by other windows in front of it, it is simply brought to the foreground of the display. If the window was hidden, it is restored in the workspace at the same position and at the same size as it existed before it was hidden. If the window has never been shown in the workspace, it is shown at a default location and size. A user may also reposition or resize a window in a workspace, without affecting the size or position of the corresponding window in another workspace.

#### B. Architecture

FIG. 1 is a block diagram of a computer system 100 in which systems consistent with the present invention may be implemented. System 100 consists of a computer 110

connected to a server 180 via a network 170. Network 170 may be a local area network (LAN), a wide area network (WAN), or the Internet. In a preferred embodiment, computer 110 is a network computer. System 100 is suitable for use in the HotJava™ Views™ user environment, although one of skill in the art will recognize that methods and apparatus consistent with the present invention may be applied to other suitable user environments. HotJava Views is a graphical user interface developed by Sun Microsystems, Inc. for network computers. It is described, for example, in a document entitled Designing the HotJava Views™ User Environment For a Network Computer, Sun Microsystems, Inc. (1998), the contents of which are hereby incorporated by reference and attached as an appendix. HotJava Views is written using, and manipulates the user interface via, the Java™ programming language. The Java programming language is described for example, in a text entitled "The Java Language Specification" by James Gosling, Bill Joy, and Guy Steele, Addison-Wesley, 1996, which is hereby incorporated by reference. Sun, Sun Microsystems, JavaSoft, the Sun Logo, Java, and HotJava Views are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Network computer 110, comprises several components that are all interconnected via a system bus 120. Bus 120 is, for example, a bi-directional system bus connecting the components of network computer 110, and contains thirty-two address lines for addressing a memory 125 and a thirty-two bit data bus for transferring data among the components. Alternatively, multiplex data/address lines may be used instead of separate data and address lines.

Network computer 110 contains a processor 115 connected to a memory 125.

Processor 115 may be microprocessor manufactured by Motorola, such as the 680X0 processor or a processor manufactured by Intel, such as the 80X86, or Pentium processor, or a SPARC™ microprocessor from Sun Microsystems, Inc. However, any other suitable microprocessor or micro-, mini-, or mainframe computer, may be utilized. Memory 125  
5 may include a RAM, a ROM, a video memory, and mass storage. The mass storage may include both fixed and removable media (e.g., magnetic, optical, or magnetic optical storage systems or other available mass storage technology).

The user typically inputs information to network computer 110 via a keyboard 130 and a pointing device, such as a mouse 135, although other input devices may be used.  
10 In return, information is conveyed to the user via display screen 140.

Computer 110 communicates with other computers on network 170 via a network interface 145, examples of which include Ethernet or dial-up telephone connections. Accordingly, computer 110 sends messages and receives data, including program code, through network 170. In accordance with the invention, one such downloaded task is the  
15 user environment application program described herein. The received code may be executed by processor 115 as it is received, and/or stored in memory 125 for later execution. Application code may be embodied in any form of computer program medium. The computer systems described herein are for purposes of example only. An embodiment of the invention may be implemented in any type of computer system or  
20 programming or processing environment.

The operating system, tasks, and data required by network computer 110 are centrally stored at server 180, and are loaded over network 170 as needed. When network computer 110 is first turned on, server 180 provides it with a boot image that includes the



operating system, such as JavaOS 1.0, developed by Sun Microsystems, Inc. Server 180 also provides to network computer 110 the HotJava Views user interface. This user interface supports multiple workspaces, such as, for example: an electronic mail workspace, an electronic calendar workspace, a workspace containing a directory of people in the organization, and a web browser workspace for viewing documents on an intranet or on the Internet. The actual set of tasks installed, and correspondingly, the workspaces available, can vary depending on how the system is configured by a systems administrator.

FIG. 2a is a representative user interface screen showing a workspace consistent with the present invention. FIG. 2a shows a selector 210, which is a vertical column on the left side of the screen that contains an icon corresponding to each workspace. Selector 210 functions as a workspace switch that allows the user to switch between the different workspaces. The user selects a workspace by clicking on one of the icons in selector 210. In response, computer 110 displays the windows contained in that workspace.

Selector 210 is created by a user environment (UE) application program that is part of HotJava Views and is provided to network computer 110 by server 180 during startup. The UE application program is responsible for creating and managing the graphical user interface used in HotJava Views. For example, the UE application program maintains in memory 125 an internal data structure that records which windows have been opened in which workspaces, along with their current location and size.

In FIG. 2a, selector 210 contains icons corresponding to multiple workspaces including Mail, Personal Organization, and other workspaces such as Workspace N. The Personal Organization workspace is active, as represented by the bold box surrounding

its icon in selector 210, and is displayed in screen area 225. The Personal Organization workspace contains two windows: one corresponding to a Notepad task (which may allow the user to create and store short messages), and one corresponding to a Schedule task (which may show activities for a particular day, week or month).

5           The user interface also contains a sliding window panel 220 that contains an icon for every window in every workspace, whether visible or hidden. As a default, this icon list is presented vertically in alphabetical order, but the user may rearrange the ordering of the icon list if desired. As shown in FIG. 2a, window panel 220 contains icons for three windows: Inbox, Notepad, and Schedule. Window panel 220 contains icons  
10           corresponding to all three windows, even though only two windows are displayed in the selected workspace.

          In one embodiment, window panel 220 is placed adjacent to selector 210. To minimize cluttering the user interface, window panel 220 is normally hidden. When hidden, its existence is indicated to the user by a thin, vertical strip along the left edge of  
15           screen 140. The user can open window panel 220 by moving the pointer to the left edge of screen 140. This causes window panel 220 to slide out from under selector 210 into the position shown in FIG. 2a.

          FIG. 2b is a representative user interface screen showing another workspace consistent with the present invention; the screen shown in FIG. 2b corresponds to the  
20           screen shown in FIG. 2a. In FIG. 2b, the Mail workspace is active, as represented by the bold box surrounding its icon in selector 210, and is displayed in screen area 225. The Mail workspace contains two windows: one corresponding to an Inbox task (which may show incoming electronic mail messages), and one corresponding to the Notepad task

shown in FIG. 2a. Again, window panel 220 contains icons corresponding to all three windows, even though only two windows are displayed in the selected workspace.

FIG. 2b also shows a drawer 230, which contains icons corresponding to tasks that can be launched. Drawer 230 contains icons corresponding to tasks such as Notepad, Spreadsheet and others such as Task X.

C. Architectural Operation

For purposes of explanation, the following description is based on the screen displays shown in FIGS. 2a and 2b. Those of skill in the art will recognize that the combinations shown herein are exemplary only.

FIG. 3 is a flow diagram of operations performed to launch the Notepad task shown in FIG. 2b in a manner consistent with the present invention. The process begins when system 100 receives a command from the user selecting the Mail workspace (step 310) by clicking on its icon in selector 210. After the Mail workspace has been selected, system 100 receives a command from the user to launch the Notepad task (step 320). In one embodiment, the user does so by using mouse 135 to move the pointer to the bottom of the screen; this opens drawer 230, which contains icons for various tasks that may be launched. The user points to the icon corresponding to Notepad and clicks the mouse button to launch the task.

In response to the user's command to launch the Notepad task, processor 115 checks to determine whether the Notepad task is already displayed in a window in the Mail workspace (step 330). If so (as in this example), processor 115 causes the corresponding window to be brought to the foreground of the display (step 360), and the procedure ends. If the Notepad task had not already been displayed in a window in the Mail workspace, processor 115 would cause a window containing the Notepad task to be

created and shown on display screen 140 (step 340). In addition, processor 115 would cause an icon to be added to window panel 220, corresponding to the Notepad task that was just launched (step 350).

FIG. 4 is a flow diagram of operations performed to hide the Notepad window in the Mail workspace, and assumes that the Notepad window is displayed in both the Mail and Personal Organization workspaces. The procedure begins when system 100 receives a command from the user selecting the Mail workspace, which may be done by using selector 210 (step 410). In response, processor 115 displays on screen 140 all windows shown in that workspace. System 100 then receives a command from the user that indicates that the user wishes to hide the Notepad window (step 420); this may be accomplished by having the user click on a designated button in the frame of the Notepad window. In response to this command, processor 115 stores the size and position of the Notepad window in memory 125 (step 430). This is done so that if the user subsequently desires to view the window, it can be displayed at the same position and at the same size as it previously existed in the workspace. Finally, processor 115 causes the Notepad window to be removed from the display of the Mail workspace (step 440).

Importantly, although the window is hidden in the Mail workspace, any corresponding version of the window that is open in another workspace (such as in the Personal Organization workspace shown in FIG. 2a) remains visible in that workspace. Moreover, although the Notepad window is hidden in the current workspace, the icon representing that window remains in window panel 220. This is necessary so that the user can later show the Notepad window in the Mail workspace by using window panel 220.

FIG. 5 is a flow diagram of operations performed to show a hidden or obscured window, and assumes that the Notepad window is hidden in both the Mail and Personal Organization workspaces. The procedure begins when system 100 receives a command from the user selecting a workspace, which may be done by using selector 210 (step 505).  
5 For example, the user may select the Mail workspace. System 100 then receives a command from the user to access window panel 220, which may be accomplished by the user pointing to the left edge of screen 140 (step 510). In response to the user's pointing, window panel 220 slides open next to selector 210 (step 515).

Once window panel 220 is displayed, system 100 receives a command from the  
10 user to display the window; the user may issue this command by using mouse 135 to click on an icon representing the Notepad window (step 520). In response, processor 115 first checks to determine whether the Notepad window is already being displayed in the current workspace (step 525). If it is, processor 115 causes the Notepad window to be moved to the foreground of display 140 (step 530). This allows the user to bring to the  
15 foreground a window that already exists in the current workspace, but has been obscured or hidden by other windows.

If the selected window is not already displayed in the current workspace (as in this example), processor 115 checks to determine whether the selected window has previously been displayed in the current workspace (step 535). If so, processor 115 recalls from  
20 memory 125 the size and position of the window as it was previously displayed (step 540). If not, a default size and position are used for the window. Next, processor 115 causes the Notepad window to be displayed in the Mail workspace (step 545), using either stored or default parameters. Importantly, although the window may be displayed

in the Mail workspace, it does not affect the display (or lack thereof) of a corresponding window in another workspace, such as the Personal Organization workspace.

FIG. 6 is a flow diagram of operations performed to terminate the Notepad window from the Personal Organization workspace, and assumes that the Notepad window is displayed in both the Mail and Personal Organization workspaces. The procedure begins when system 100 receives a command from the user selecting the Personal Organization workspace (step 610). If the Personal Organization workspace did not contain the window, an appropriate workspace could be selected by using selector 210. System 100 then receives a command from the user to terminate the Notepad window (step 620); the user may issue this command by clicking on a designated button on the Notepad window. Alternatively, the user may terminate a window directly from window panel 220 by pointing at the icon representing that window and using a pop-up menu that contains a "terminate" command. In response, processor 115 removes the Notepad window from the display of all workspaces (step 630). Processor 115 also causes the icon representing the Notepad window to be removed from window panel 220 (step 640), and the procedure ends.

FIG. 7 is a flow diagram of steps performed to reposition or resize the Notepad window in the Mail workspace, and assumes that the Notepad window is displayed in both the Mail and Personal Organization workspaces. The procedure begins when system 100 receives a command from the user selecting the Mail workspace, which may be done by using selector 210 (step 710). In response, processor 115 causes to be displayed on screen 140 all windows contained in that workspace. System 100 then receives a command from the user selecting the Notepad window for repositioning or resizing (step 720), which may be done by using mouse 135. System 100 then receives a command

from the user to reposition or resize the Notepad window (step 730), again by using mouse 135. In response to the user's actions, processor 115 causes the Notepad window's size or position to change on display screen 140 (step 740), and the procedure ends. Importantly, the user's actions in repositioning or resizing the Notepad window in the Mail workspace does not affect the position or size of the corresponding Notepad window in other workspaces, such as the Personal Organization workspace.

D. Conclusion

As described in detail above, methods and apparatus consistent with the present invention allow a user to easily select and manipulate windows in a user interface that supports multiple workspaces. The foregoing description of an implementation of the invention has been presented for purposes of illustration and description. For example, the described implementation includes software but the present invention may be implemented as a combination of hardware and software or in hardware alone. Modifications and variations are possible to the processes described in connection with FIGS. 3-7 in light of the above teachings or may be acquired from practicing the invention.

Although systems and methods consistent with the present invention are described as operating in the exemplary distributed system and the Java™ programming environment, one skilled in the art will appreciate that the present invention can be practiced in other systems and programming environments. Additionally, although aspects of the present invention are described as being stored in memory, one skilled in the art will appreciate that these aspects can also be stored on other types of computer-readable media, such as secondary storage devices, like hard disks, floppy disks, or CD-

ROM; a carrier wave from the Internet; or other forms of RAM or ROM. The scope of the invention is therefore defined by the claims and their equivalents.



What is claimed is:

1. A method for managing the display of windows corresponding to tasks executable by a computer, the method comprising:

5 providing multiple workspaces, each workspace being capable of displaying multiple windows corresponding to executing tasks;

displaying a window panel including icons corresponding to the executing tasks; and

10 permitting shared access to a window of the executing tasks upon selection of a corresponding icon displayed in the window panel.

2. The method of claim 1, further comprising adding to the window panel an icon for each task opened into a window.

15 3. The method of claim 1, further comprising:  
receiving a command from a user to launch a task; creating a window for the task; and  
adding an icon representing the window to the window panel.

20 4. The method of claim 1, further comprising:  
receiving a command from a user to hide a window in a selected workspace;  
removing the window only from the selected workspace; and  
maintaining an icon representing the window in the window panel.

5. The method of claim 4, further comprising recording a size and location for the window in the selected workspace.

5 6. The method of claim 1, further comprising:  
receiving a command from a user, in a particular workspace, selecting an icon in the window panel; and  
displaying in the foreground of the workspace a window corresponding to the selected icon.

10 7. The method of claim 1, further comprising the steps of:  
receiving a command from a user to terminate a window;  
removing the window from each workspace in which it is displayed; and  
deleting from the window panel an icon corresponding to the terminated window.

15 8. The method of claim 1, further comprising the steps of:  
displaying a window in more than one workspace; and  
resizing the window in one workspace without affecting the size of the window in another workspace.

20 9. The method of claim 1, further comprising the steps of:  
displaying a window in more than one workspace; and  
repositioning the window in one workspace without affecting the position of the window in another workspace.

10. The method of claim 1, further comprising the steps of:  
hiding the window panel behind a workspace switch;  
receiving a command from the user selecting the workspace switch; and  
5 displaying the window panel in response to the user command selecting the switch.
11. A computer-readable medium containing instructions for managing the display of windows corresponding to tasks executable by a computer, by:  
10 providing multiple workspaces, each workspace being capable of displaying multiple windows corresponding to executing tasks;  
displaying a window panel including icons corresponding to the executing tasks; and  
permitting shared access to a window of the executing tasks upon  
15 selection of a corresponding icon displayed in the window panel.
12. The computer-readable medium of claim 11, further comprising adding to the window panel an icon for each task opened into a window.
- 20 13. The computer-readable medium of claim 11, further comprising:  
receiving a command from a user to launch a task;  
creating a window for the task; and  
adding an icon representing the window to the window panel.

14. The computer-readable medium of claim 11, further comprising:  
receiving a command from a user to hide a window in a selected  
workspace;  
removing the window only from the selected workspace; and  
5 maintaining an icon representing the window in the window panel.
15. The computer-readable medium of claim 14, further comprising recording  
a size and location for the window in the selected workspace.
- 10 16. The computer-readable medium of claim 11, further comprising:  
receiving a command from a user, in a particular workspace, selecting an  
icon in the window panel; and  
displaying in the foreground of the workspace a window corresponding  
to the selected icon.
- 15 17. The computer-readable medium of claim 11, further comprising the steps  
of:  
receiving a command from a user to terminate a window;  
removing the window from each workspace in which it is displayed; and  
20 deleting from the window panel an icon corresponding to the terminated  
window.
18. The computer-readable medium of claim 11, further comprising the steps  
of:

displaying a window in more than one workspace; and  
resizing the window in one workspace without affecting the size of the  
window in another workspace.

5           19.    The computer-readable medium of claim 11, further comprising the steps  
            of:

            displaying a window in more than one workspace; and  
            repositioning the window in one workspace without affecting the position  
            of the window in another workspace.

10

            20.    The computer-readable medium of claim 11, further comprising the steps  
            of:

            hiding the window panel behind a workspace switch;  
            receiving a command from the user selecting the workspace switch; and  
15           displaying the window panel in response to the user command selecting  
            the switch.

            21.    An apparatus for managing the display of windows corresponding to tasks  
            executable by a computer, comprising:

20           a memory having program instructions, and  
            a processor configured to use the program instructions to:  
            provide multiple workspaces, each workspace being capable of displaying  
            multiple windows corresponding to executing tasks;

display a window panel including icons corresponding to the executing tasks; and

permit shared access to a window of the executing tasks upon selection of a corresponding icon displayed in the window panel.

5

22. The apparatus of claim 21, wherein the processor is further configured to add to the window panel an icon for each task opened into a window.

23. The apparatus of claim 21, wherein the processor is further configured to:  
10 receive a command from a user to launch a task;  
create a window for the task; and  
add an icon representing the window to the window panel.

24. The apparatus of claim 21, wherein the processor is further configured to:  
15 receive a command from a user to hide a window in a selected workspace;  
remove the window only from the selected workspace; and  
maintain an icon representing the window in the window panel.

25. The apparatus of claim 24, wherein the processor is further configured to  
20 record a size and location for the window in the selected workspace.

26. The apparatus of claim 21, wherein the processor is further configured to:  
receive a command from a user, in a particular workspace,  
selecting an icon in the window panel; and

display in the foreground of the workspace a window corresponding to the selected icon.

- 5           27.    The apparatus of claim 21, wherein the processor is further configured to:  
            receive a command from a user to terminate a window;  
            remove the window from each workspace in which it is displayed; and  
            delete from the window panel an icon corresponding to the terminated  
            window.
- 10           28.    The apparatus of claim 21, wherein the processor is further configured to:  
            display a window in more than one workspace; and  
            resize the window in one workspace without affecting the size of the  
            window in another workspace.
- 15           29.    The apparatus of claim 21, wherein the processor is further configured to:  
            display a window in more than one workspace; and  
            reposition the window in one workspace without affecting the position of  
            the window in another workspace.
- 20           30.    The apparatus of claim 21, wherein the processor is further configured to:  
            hide the window panel behind a workspace switch;  
            receive a command from the user selecting the workspace switch; and  
            display the window panel in response to the user command selecting the  
            switch.

31. An apparatus for managing the display of windows corresponding to tasks  
executable by a computer, comprising:

means for providing multiple workspaces, each workspace being capable  
of displaying multiple windows corresponding to executing tasks;

5 means for displaying a window panel including icons corresponding to the  
executing tasks; and

means for permitting shared access to a window of the executing tasks  
upon selection of a corresponding icon displayed in the window panel.

10 32. The apparatus of claim 31, further comprising means for adding to the  
window panel an icon for each task opened into a window.

33. The apparatus of claim 31, further comprising:

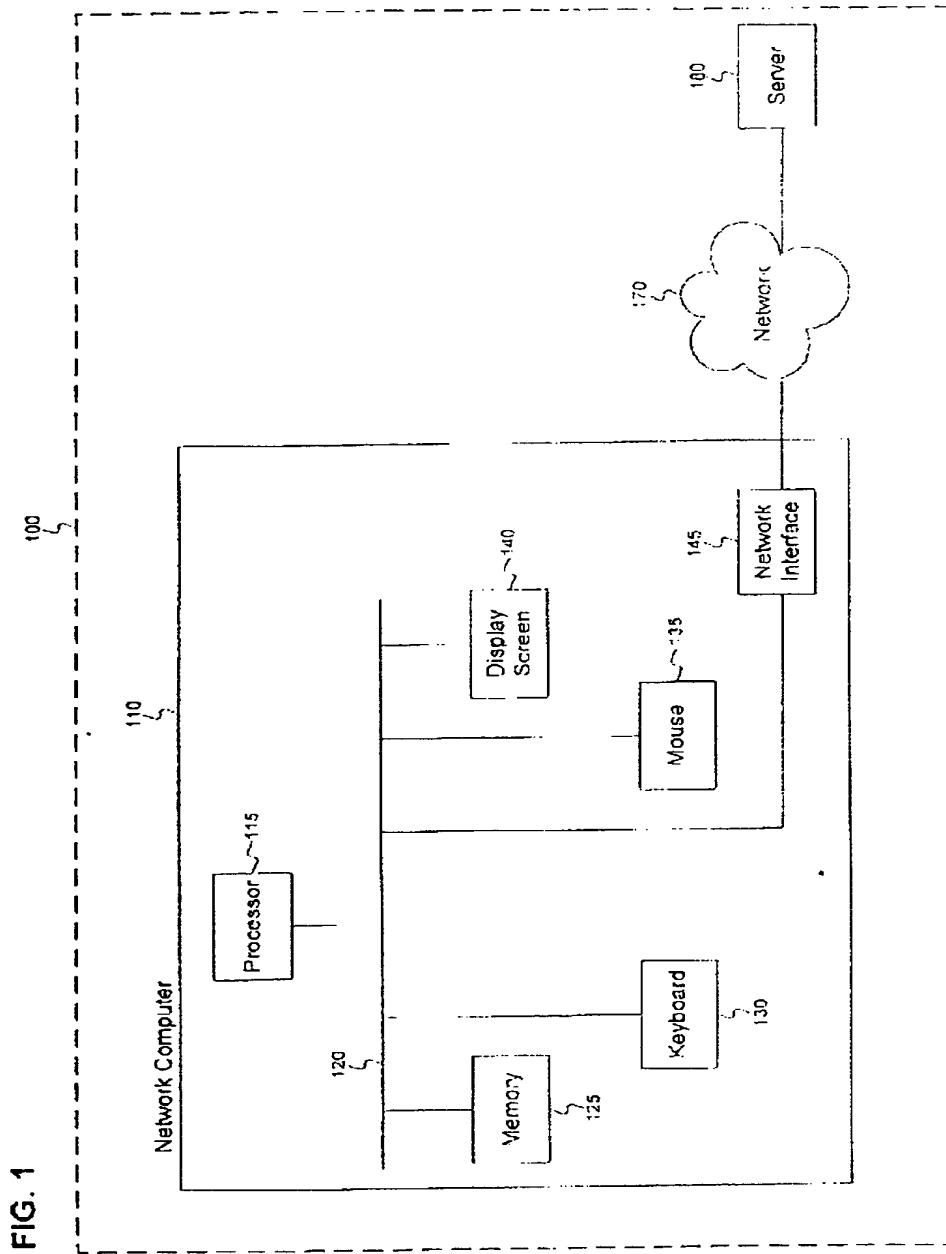
means for receiving a command from a user to launch a task;

15 means for creating a window for the task; and

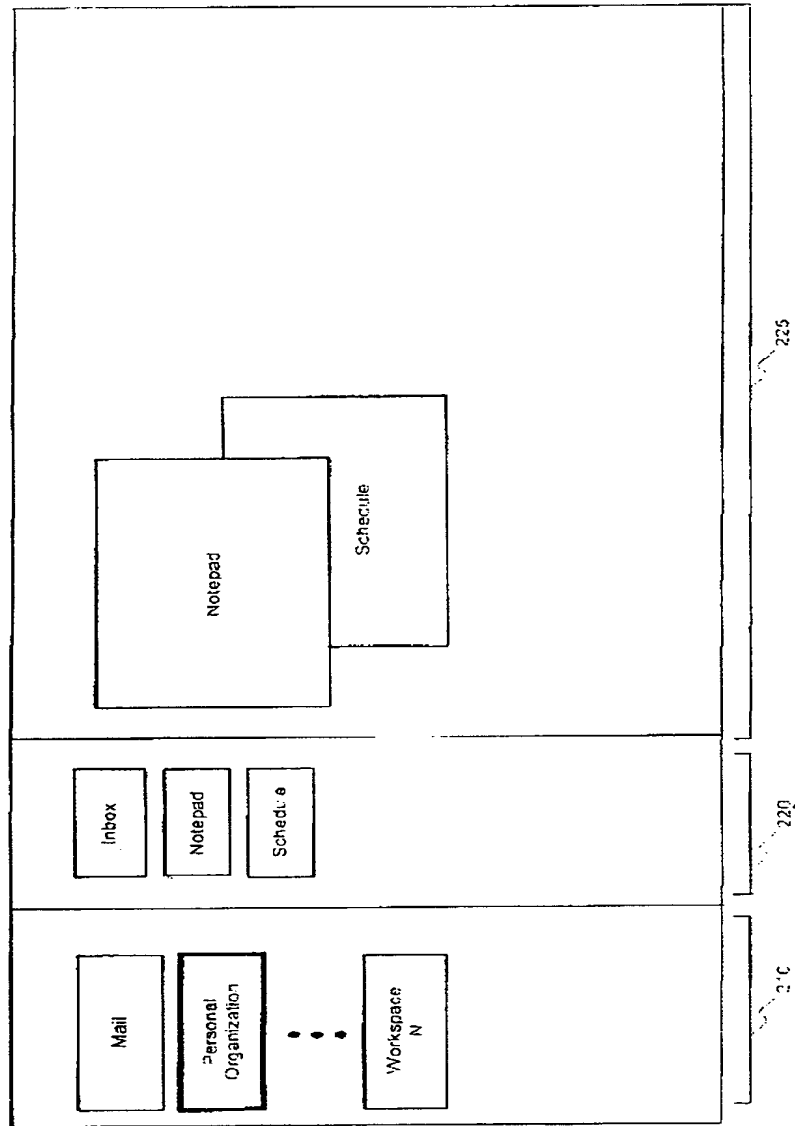
means for adding an icon representing the window to the window panel.



【 図 1 】



【 図 2 】



【図3】

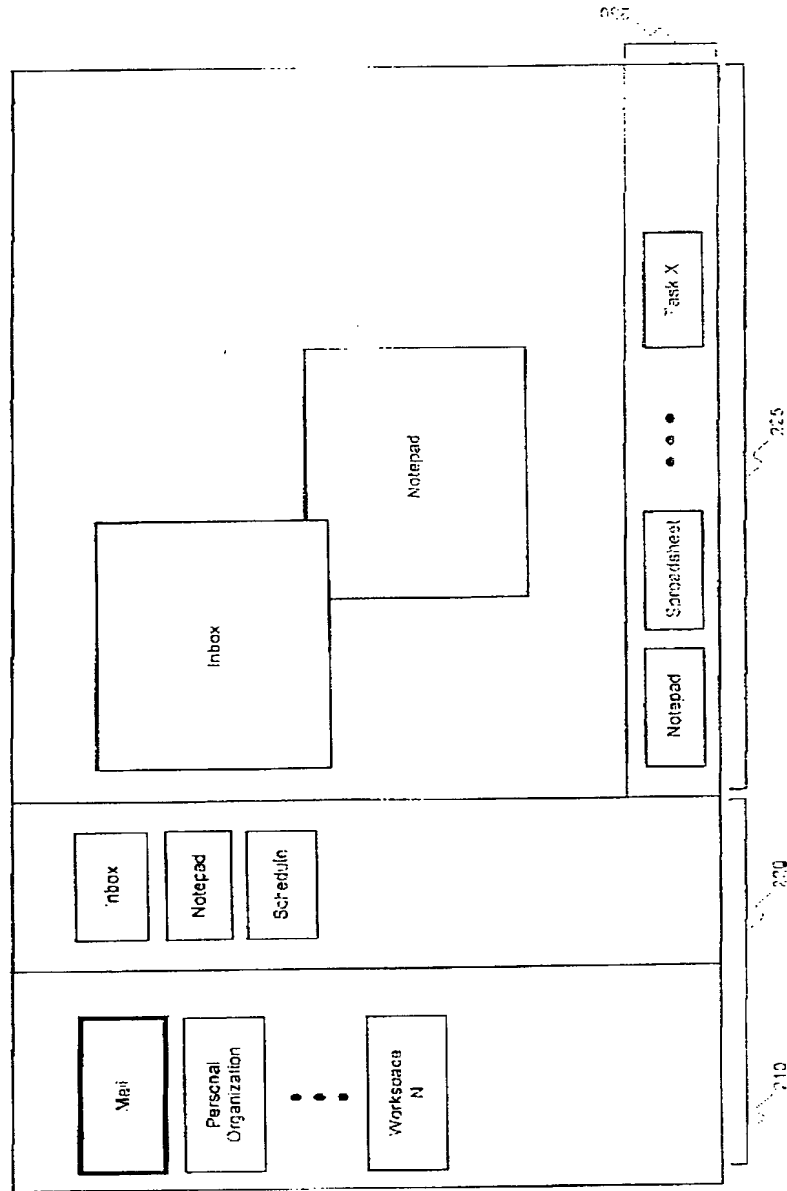
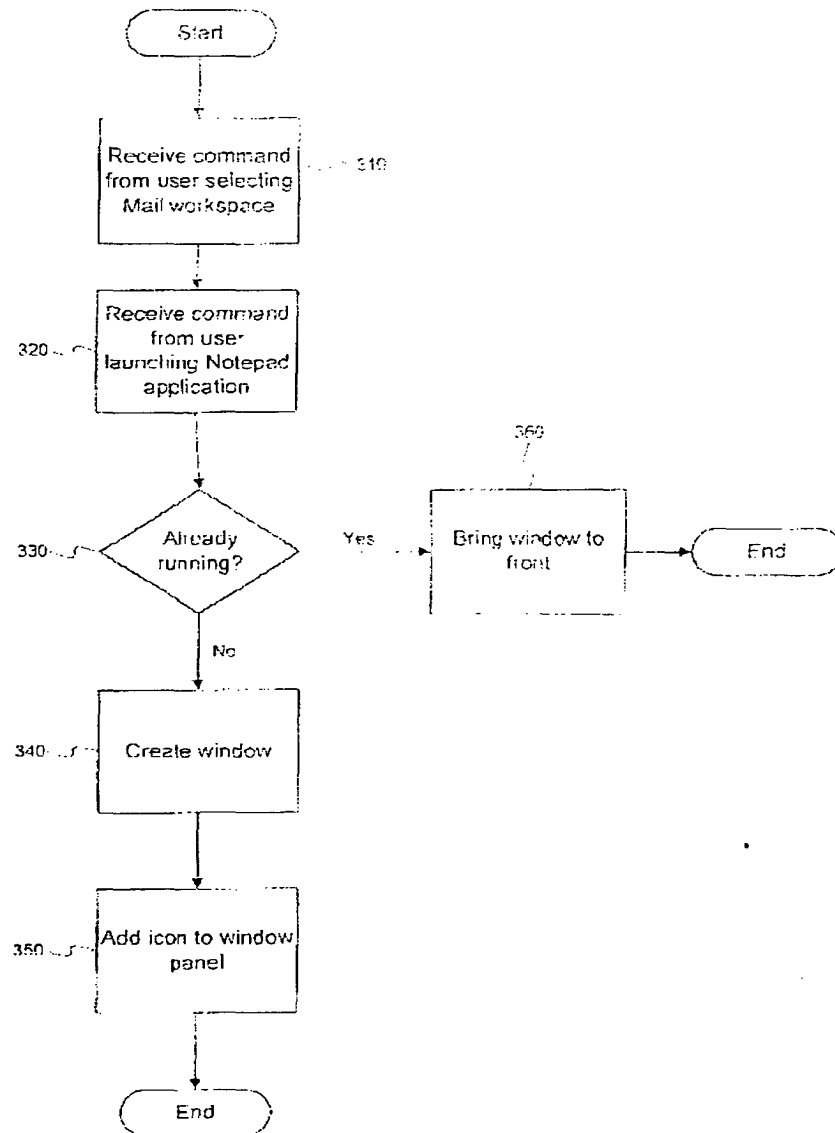


FIG. 2b

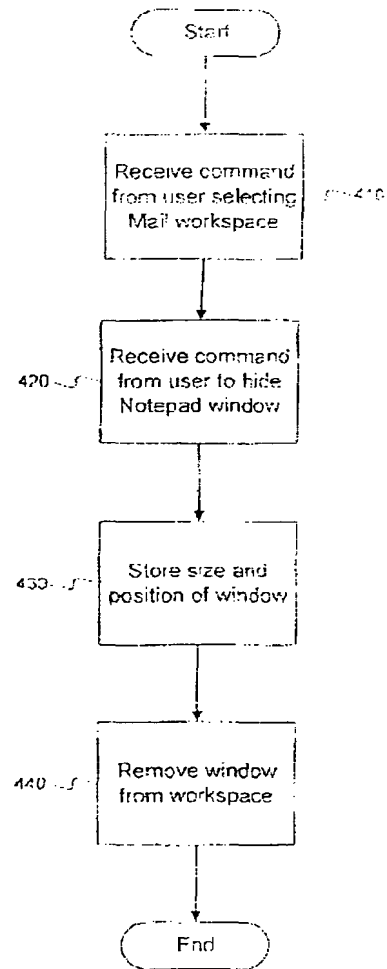
【図4】

FIG. 3



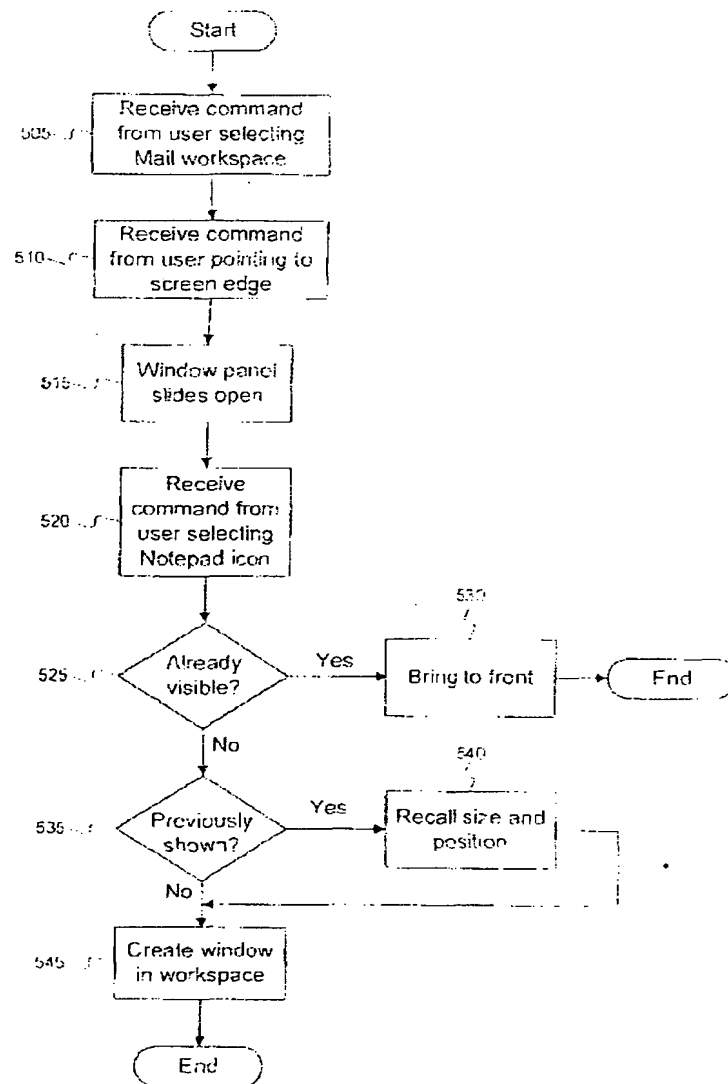
[ 図 5 ]

FIG. 4



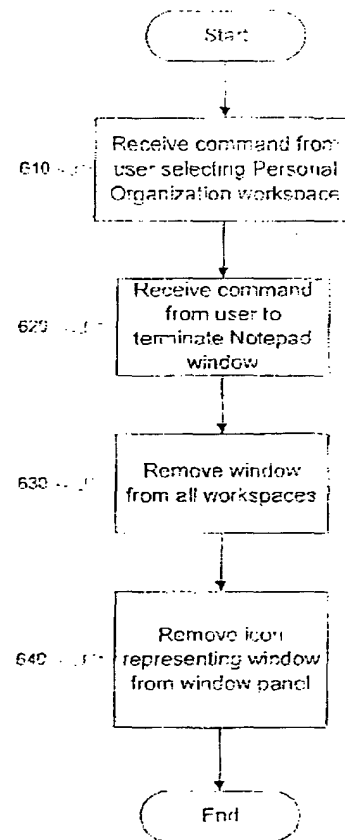
【 図 6 】

FIG. 5



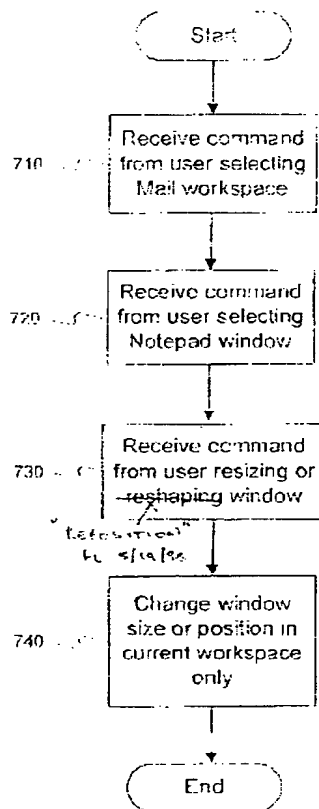
【図7】

FIG. 6



【図8】

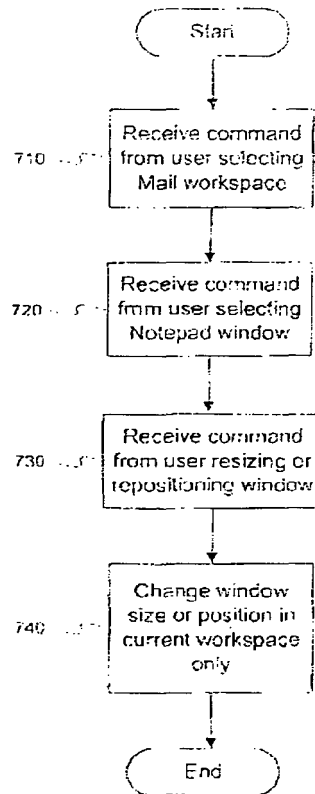
FIG. 7





【図9】

FIG. 7



**ABSTRACT OF THE DISCLOSURE**

A system employs a sliding window panel that contains icons representing every task that has been opened into a window, regardless of the workspace in which it exists.

A user may use the sliding window panel to launch, terminate, hide, or resize windows  
5 in the workspaces.